DOS 5.0

COMPLETE

MS-DOS 5.0
ScreenSaver
INCLUDED ON COMPANION DISK

A practical user's guide to learning and using the new MS-DOS 5.0

Includes ready-to-use companion disk

- Learn the " ins and outs" of using the new MS-DOS 5.0.
- Boost your productivity with these practical techniques.
- Discover ways to solve your own DOS problems.
- Save valuable time with the ready-to-run companion disk.
- Browse the extensive MS-DOS reference section.

By Manfred and Helmut Tornsdorf

Abacus
A Data Becker Book

DOS 5.0

COMPLETE

Manfred and Helmut Tornsdorf

Abacus
A Data Becker Book

Copyright © 1991, 1992

Abacus
5370 52nd Street, S.E.
Grand Rapids, MI 49512

Copyright © 1991, 1992

DATA BECKER GmbH
Merowingerstrasse 30
4000 Duesseldorf, Germany

Editors: George Miller, Robbin Markley,
Louise Benzer, Scott Slaughter,
Gene Traas, Jim D'Haem

This book is copyrighted. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of Abacus or Data Becker, GmbH.

Every effort has been made to ensure complete and accurate information concerning the material presented in this book. However, Abacus can neither guarantee nor be held legally responsible for any mistakes in printing or faulty instructions contained in this book. The authors always appreciate receiving notice of any errors or misprints.

This book contains trade names and trademarks of many companies and products. Any mention of these names or trademarks in this book are not intended to either convey endorsement or other associations with this book.

PC-DOS, IBM PC, XT, AT, PS/2, OS/2 and PC-BASIC are trademarks or registered trademarks of International Business Machines Corporation. Microsoft Works, Microsoft Quick C, Microsoft Windows, MS-DOS, QuickBASIC and QBASIC are trademarks or registered trademarks of Microsoft Corporation.

Tornsdorf, H. (Helmut), 1951-
DOS 5.0 complete / H. Tornsdorf, M. Tornsdorf.
p. cm.
"A Data Becker book."
Includes index.
ISBN 1-55755-109-X : $34.95
1. PC DOS (Computer operating system) 2. MS-DOS (Computer operating system) I. Tornsdorf, M. (Manfred), 1960-
II. Title.
QA76.76.063T673 1991 91-7351
005.4'46--dc20 CIP

Printed in U.S.A.
10 9 8 7 6 5 4 3

Foreword

Microsoft DOS 5.0 expands upon and improves the earlier versions of the operating system for IBM compatible personal computers. With this new version of MS-DOS you can work with more than one program at the same time, within certain limitations. Also, the new user interface, MS-DOS Shell, has a great deal to offer users.

Besides the numerous improvements to previous versions of DOS, MS-DOS 5.0 achieves two very important things:

- MS-DOS 5.0 doesn't require as much memory as earlier versions of DOS for computers that have at least 1 Meg of memory. This means you will have more memory available for application programs. MS-DOS 5.0 users will no longer sacrifice memory for enhanced DOS convenience and performance.
- The new MS-DOS Shell of MS-DOS 5.0 closes the gap between the user interface of the MS-DOS operating system and Windows 3.0 by providing a "character oriented" user interface.

However, any program (and MS-DOS 5.0 itself is a computer program) is only as good as its documentation and the ability of the user to understand its features.

You'll find that *DOS 5.0 Complete* is a combination of technical information and understandable, practical, problem oriented descriptions. Regardless of your knowledge level, you'll find this the ideal source of information for MS-DOS 5.0. *DOS 5.0 Complete* is a valuable reference book for solving problems and a source of excellent tips and tricks that will enable you to utilize the full power of MS-DOS 5.0.

About this book

DOS 5.0 Complete is a combination of an introduction and reference guide to MS-DOS 5.0 for beginners and advanced users. We'll learn the ins and outs of this new operating system. During your initial learning sessions, you'll need information about the functions and commands of the operating system. Once you have some experience with DOS 5.0, you won't need a reference for the syntax and explanation of commands.

The problem oriented section

DOS 5.0 Complete provides a comprehensive introduction to the most important components of MS-DOS for users who want to learn MS-DOS 5.0 quickly. This section provides important information and tips and tricks that you can use immediately.

The reference section

The MS-DOS commands are listed alphabetically and described in *Syntax*, *Options* and *Description* sections. Certain characters and notations used in these sections have the following meaning:

[ ]	Parameters enclosed in square brackets are optional.
|	Alternative parameters are separated by a vertical bar. Only one of the parameters separated by a vertical bar can be entered in the command.
	Parameter names are indicated by italic type, such as *filename*. This helps you distinguish parameters and commands from one another.
path	Specifies the hierarchical order of the directories separated by a backslash.
filename	Specifies the name of the file and may include an extension.
pathname	Specifies the *path* followed by the *filename*.
*	Any character or combination of characters may be replaced by an asterisk (*). This wildcard character can be located in any position of the filename or extension, but wildcards are permitted only when explicitly noted.
?	Any single character may be replaced by a question mark (?). This wildcard character can be located in any position of the filename or extension, but wildcards are permitted only when explicitly noted.

Options are listed after each command's *Syntax* for easy reference.

Icons used in this book

In this book you'll see a number of icons, which will help you locate information quickly:

➡ The "Warning" icon indicates the possibility of errors and other problems. Read the accompanying paragraph before invoking the command or function.

✗ The "Error" icon indicates typical errors that may occur. Reading the accompanying paragraph may help you avoid these errors. If an error does occur, this text often lists a solution to the problem.

✎ The "Note" icon provides additional information about the current topic, or refer you to other locations where you can find more information.

☞ The "Hint" icon indicates tips and tricks for using the corresponding command or function.

Getting the most out of *DOS 5.0 Complete*

- If you're new to MS-DOS 5.0, read Chapter 1, "Introduction to MS-DOS 5.0". If you still haven't installed MS-DOS 5.0, you'll find the necessary information in your MS-DOS 5.0 User's Manual.
- If you need information about a new subject, refer to the specific section of *DOS 5.0 Complete* that describes that particular subject. All the basic information is located here. Note any references and turn to the reference section for additional information.
- To find the important options of a command, with which you're familiar, skim the appropriate chapter. Pay attention to the corresponding captions.
- If you're familiar with a basic subject, but need special details, use the reference section. This section contains an overview that can be used to quickly find the necessary information. In addition, the comprehensive index also helps you locate information quickly.

Different versions of DOS

DOS 5.0 Complete covers both MS-DOS and the IBM version called PC-DOS. Since IBM is a licensed version of MS-DOS, the differences between the two versions are insignificant. In the following sections we'll refer only to MS-DOS. Any differences will be pointed out in the appropriate sections.

We hope that this book comes as close as possible to your needs and expectations.

Helmut Tornsdorf, Manfred Tornsdorf and company

Table of Contents

Foreword .... **iii**

1. Introduction To MS-DOS 5.0 .... **1**
1.1 Starting MS-DOS 1
1.2 Using The Command Line 5
1.2.1 Displaying files 7
1.2.2 Copying files 12
1.2.3 Looking at files 15
1.2.4 Starting and quitting programs 16
1.2.5 Deleting files 21
1.2.6 Formatting diskettes 22
1.2.7 Copying diskettes 23
1.3 Introduction To The MS-DOS Shell 25
1.3.1 Displaying files with the MS-DOS Shell 25
1.3.2 Copying files with the MS-DOS Shell 28
1.3.3 Using the MS-DOS Shell to view files 32
1.3.4 Running programs from the MS-DOS Shell 35
1.3.5 Deleting files using the MS-DOS Shell 38
1.3.6 Formatting diskettes with the MS-DOS Shell 39
1.3.7 Copying diskettes with the MS-DOS Shell 41
1.4 What You Should Know About MS-DOS 43
1.4.1 What is MS-DOS? 43
1.4.2 The command line and the MS-DOS Shell 44
1.4.3 What are files, directories etc 45
1.4.4 What to do if you encounter problems 48

2. Working with MS-DOS 5.0 .... **51**
2.1 Setting Up MS-DOS 51
2.2 Starting MS-DOS 62
2.3 Using The Keyboard 65
2.4 Entering And Repeating Command Lines 67
2.5 Entering And Correcting Commands With DOSKEY 69
2.6 Using DOS Commands And Files 79
2.7 Working With Disk Drives And Hard Drives 87
2.8 Starting Application Programs 91

3. Storage Devices **93**
3.1 Working With Storage Media 93
3.2 Different Storage Devices 95
3.3 Formatting Diskettes 101
3.4 Formatting Hard Drives 111
3.5 Labeling Storage Media 129
3.6 Displaying The Contents 131
3.7 Copying Diskettes 136

3.8 Comparing Diskettes........ 139
3.9 Checking Storage Media........ 141
3.10 Redirecting Drive Access........ 146
3.11 Creating A RAM Disk........ 152
3.12 Speeding Up Your Hard Drive Access........ 157
3.13 MS-DOS Internals........ 161

4. Order Through Directories........ 173
4.1 MS-DOS And Subdirectories........ 175
4.2 Setting Up A Directory........ 180
4.3 Choosing A Directory........ 182
4.4 Search Paths For Commands And Files........ 185
4.5 Displaying And Printing the Directory........ 193
4.6 Displaying And Printing All Directories........ 201
4.7 Renaming Directories........ 205
4.8 Removing Directories........ 207
4.9 Making A Directory Into A Pseudo Drive........ 209

5. Handling Files........ 211
5.1 Naming Files........ 211
5.2 Creating Files........ 215
5.3 Copying Files........ 218
5.4 Comparing Files........ 225
5.5 Viewing File Contents........ 231
5.6 Printing Files........ 233
5.7 Deleting Files........ 236
5.8 Renaming Files........ 238
5.9 Special Features Of Files........ 239
5.10 Checking Files........ 244
5.11 Searching For Files........ 247
5.12 Searching Contents Of Files........ 248
5.13 Working Faster With Files........ 249

6. Backing Up And Recovering Data........ 251
6.1 Why Backup Your Data?........ 251
6.2 Backing Up The Hard Drive........ 253
6.3 Restoring All Files........ 255
6.4 Backing Up Selected Files........ 257
6.5 Restoring Selected Files........ 263
6.6 Automatic Backup........ 265
6.7 FORMAT And DEL Or ERASE........ 268
6.8 Recovering From An Accidental Format........ 270
6.9 Recovering Files From Defective Storage Media........ 272
6.10 Restoring Deleted Files........ 275

7. Application Programs 279
7.1 Installing Programs 281
7.2 The Command Interpreter 290
7.3 Starting Programs From The MS-DOS Shell 293
7.4 Starting Programs With Files Or Parameters 305
7.5 Quitting Programs 308
7.6 Avoiding Problems With Programs 309

8. Data Input And Output 315
8.1 Setting Screen Output 315
8.2 Redirecting Input And Output 327
8.3 Linking Commands 333

9. System Settings 335
9.1 Setting The Time And Date 335
9.2 Customizing The System Prompt 341
9.3 Specifying The Command Interpreter 345
9.4 Using The Command Interpreter 349
9.5 Environment Variables 352
9.6 Special System Settings 355
9.7 Setting The Number Of Buffers And Files 357

10. Peripheral Devices 359
10.1 Installing A Printer 359
10.2 Setting Serial Ports 362
10.3 Setting The Display Mode 364
10.4 Screen Control System And Keyboard Actions 366
10.5 Modifying The Keyboard 375
10.6 Defining New Disk Drives 378

11. Memory Management 381
11.1 Background 381
11.2 Conventional And Extended Memory 384
11.3 Extended Memory 387
11.4 Expanded Memory 390
11.5 High Memory 393
11.6 Upper Memory 395
11.7 Requirements For Using Upper Memory 397
11.8 What Is XMS Memory? 399
11.9 Customizing Memory To Your Requirements 400
11.10 Partitioning Memory 403

12. The EDLIN Editor 405
12.1 Editing A Batch File 405
12.1.1 Editing lines with EDLIN 406
12.1.2 Inserting lines with EDLIN 411
12.2 File Maintenance With EDLIN 413
12.2.1 Editing files created with COPY CON 413
12.2.2 Creating a new file with EDLIN 413
12.2.3 Printing with EDLIN 414
12.3 Special Capabilities Of EDLIN 415
12.3.1 Replacing characters and words with EDLIN 415
12.3.2 Moving lines with EDLIN 416
12.3.3 Copying lines with EDLIN 416
12.3.4 Displaying and deleting lines 418
12.4 Review 419

13. Using The MS-DOS Editor 421
13.1 Starting The Editor And Loading A File 425
13.2 Saving A File 430
13.3 Entering And Editing Text 432
13.4 Working With Blocks Of Text 436
13.5 Edit Keys 439
13.6 Moving The Cursor 441
13.7 The Edit Menus 442

14. Batch Files 449
14.1 Batch Files And Programs 450
14.2 Creating And Running Batch Files 453
14.3 Using Batch Files 456
14.4 Displaying And Modifying Batch Files 458
14.5 Batch File Commands 461
14.6 Batch Files And Parameters 477
14.7 Batch Files And Environment Variables 480
14.8 Forbidden Characters In Batch Files 482
14.9 DOSKEY And Batch Files 483

15. The MS-DOS Shell 487
15.1 Advantages Of The MS-DOS Shell 489
15.1.1 MS-DOS Shell Basics 489
15.1.2 Opening the MS-DOS Shell 491
15.2 MS-DOS Shell Fundamentals 495
15.2.1 Working in an area 495
15.2.2 Using the pull-down menus 496
15.2.3 Selecting files and other elements 499
15.3 Working With Groups And Programs 505
15.3.1 Groups and programs 505
15.3.2 Starting programs 506

15.3.3 Selecting groups 508
15.3.4 Working with more than one program 508
15.3.5 Special features for more than one program 514
15.3.6 Installing and changing programs 516
15.3.7 Installing programs with parameters 525
15.3.8 Installing program groups 527
15.3.9 Programs in the program list area 530
15.3.10 Protecting groups and programs 533
15.4 Working With Files And Directories 534
15.4.1 The screen of the Directory Tree/file list area 534
15.5 The MS-DOS Shell Menus 536
15.5.1 The File menu for the program list area 536
15.5.2 The File menu for the file list area 538
15.5.3 The Options menu 548
15.5.4 The View menu 555
15.5.5 The Tree menu 558
15.5.6 The Help menu 559

16. Using QBasic 561
16.1 Working With QBasic 561
16.1.1 Special features of QBasic 562
16.1.2 Starting QBasic 563
16.1.3 Running QBasic 565
16.2 The QBasic Menus 568
16.2.1 Loading and saving programs 580
16.3 Commands And Functions 583
16.3.1 Using variables for simple calculations 583
16.3.2 Input and Output of data 585
16.3.3 Commands and functions 587
16.3.4 Program control 588
16.3.5 Using subroutines 591
16.4 QBasic And Graphics 596
16.4.1 Different graphics modes 597
16.4.2 Basic graphics commands 597
16.4.3 Preserving your screen 598

17. MS-DOS Tips And Tricks 601
17.1 An Instant ASCII Table 603
17.2 Searching For Files 605
17.3 Bypassing Error Messages 606
17.4 Applying Commands 607
17.5 Screen Header Using Date And Path 609
17.6 Increasing Environment Space 610
17.7 Returning To A Directory 611
17.8 Quick Directory Changes 612
17.9 Saving Disk Space With GO.BAT 614

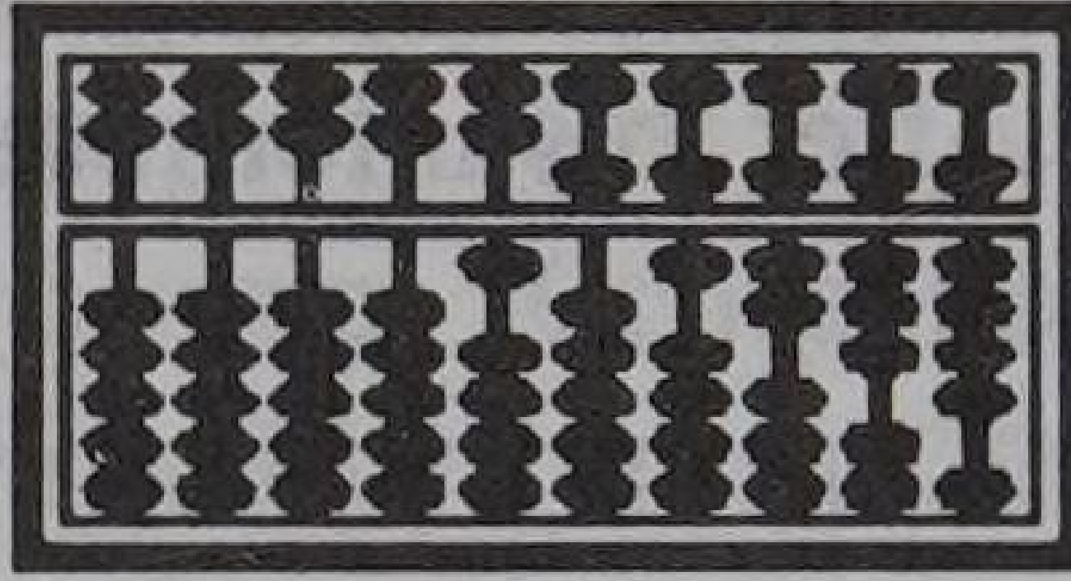
17.10 Using The <F11> And <F12> Keys 616
17.11 Page By Page Directory Display 617
17.12 Protecting Directories 618
17.13 Cancelling Disk Formatting 619
17.14 Testing Files For Readability 620
17.15 Contiguous Files 621
17.16 Disabling <Num Lock> 622
17.17 Skipping Commands In AUTOEXEC.BAT 623
17.18 Determining Disk Space 625
17.19 Fast Keyboard 627
17.20 Expanding The Path Specification 628
17.21 Copying Groups Of Files With GCOPY 629
17.22 Excluding Files From Commands 630
17.23 Warning Beep In Batch Files 634
17.24 Customizing PAUSE 635
17.25 Trace Mode For Batch Files 636
17.26 Sorting Files In A Path 638
17.27 Keyboard Shortcuts - SETFUNC0 640
17.28 Displaying Function Key Assignments 643
17.29 Changing Directories 644
17.30 Deleting Protected Files 645
17.31 Batch File Prompts 647
17.32 Displaying Subdirectories 649
17.33 Simplifying Access To A Directory 650
17.34 Directory Sequence And Search Path 651
17.35 Copying Different Diskette Formats 652
17.36 Multiple Versions Of MS-DOS 653
17.37 Conserving Hard Drive Space 654

18. The MS-DOS 5.0 Commands 655
18.1 The Types Of DOS Commands 657
18.2 Command Reference 658

Appendices 787
Appendix A: Device Drivers 787
Appendix B: A Short History Of MS-DOS 802
Appendix C: ASCII Table 808
Appendix D: Country Codes 810
Appendix E: The Companion Diskette 815
Appendix F: QBasic Keywords 832
Appendix G: Quick Index 835

Index 837

Chapter



1

Introduction To MS-DOS 5.0

This chapter is designed for users who haven't worked with MS-DOS yet. We'll present the basics you'll need in order to use MS-DOS to perform all of the fundamental tasks that beginners encounter. If you already have experience working with MS-DOS, you can skip this chapter.

We've divided this chapter into two parts. One part explains how to work at the command level. The other part explains how to work with the MS-DOS Shell.

There are two requirements for working in this chapter:

1. You must have MS-DOS 5.0 installed on your computer. You can have your dealer do this for you, or you can install it using the MS-DOS diskettes included with MS-DOS 5.0. You cannot work through this chapter unless MS-DOS is properly installed on your computer.
2. Your computer should have a hard drive. Almost all computers on the market today have a hard drive. You should also have a disk drive. The companion diskette included in this book will be used in your disk drive.

☞ MS-DOS 5.0 can be installed on a computer system using only a floppy diskette. However, the examples in this chapter might not work as presented.

1.1 Starting MS-DOS

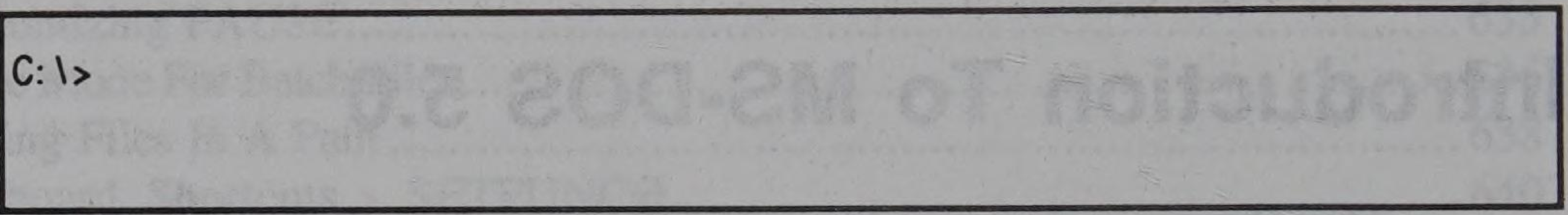
Before you start working with your computer, remove all diskettes from the disk drives and switch it on. After a few seconds, your computer is ready for operation and responds by displaying either the command line or the MS-DOS Shell, depending on the type of installation which was originally performed. (The MS-DOS 5.0 installation procedure gives you the option of always running the MS-DOS Shell on startup and includes the appropriate command in your AUTOEXEC.BAT file for doing this.)

The command line

If the command line is activated, information about the startup process is displayed on the screen. After this information is displayed, the following appears:

```
C:\>
```

This is referred to as the system prompt.

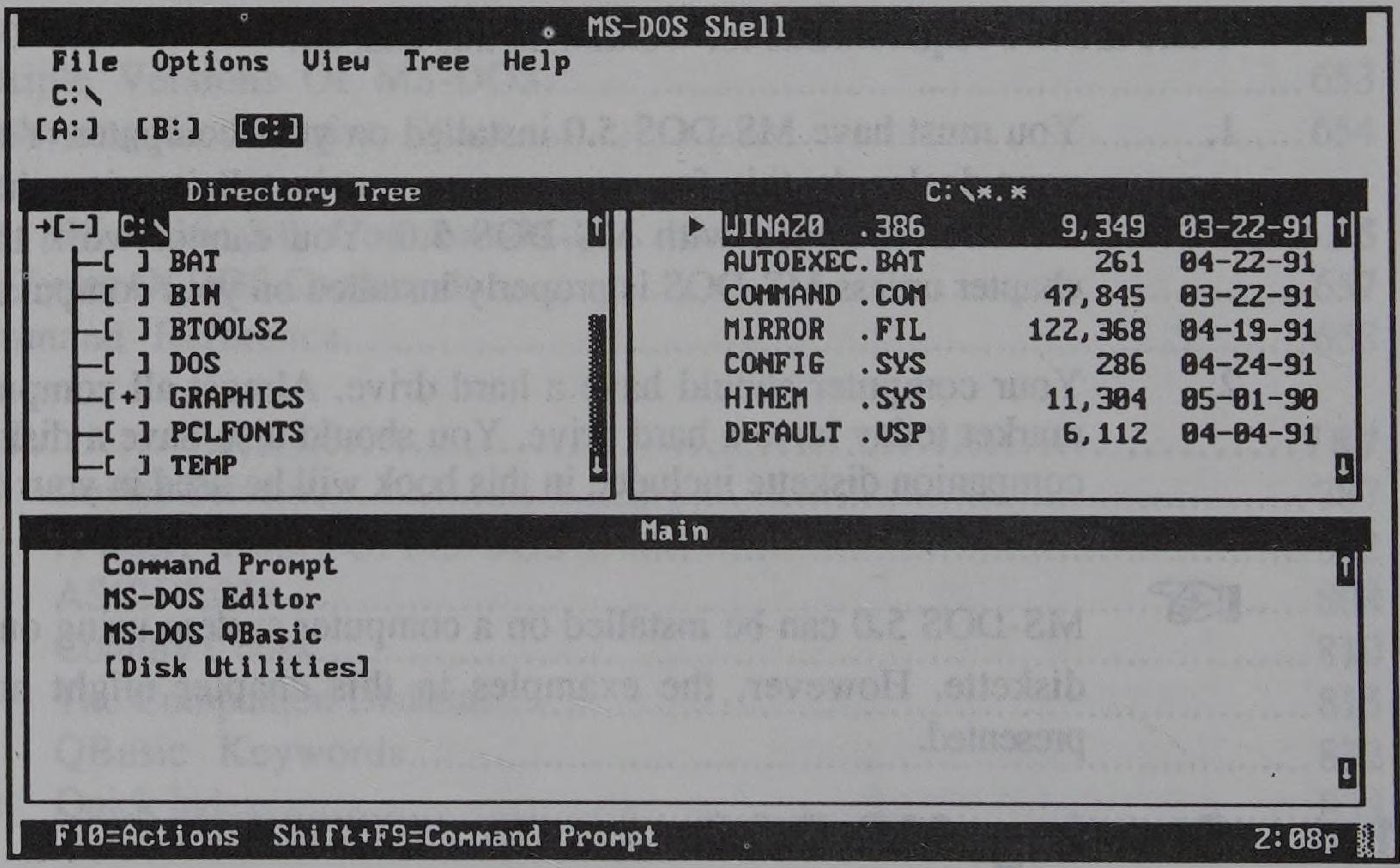


```
C:\>
```

The DOS command line

The MS-DOS Shell

As an alternative to the command line, your computer might automatically start the MS-DOS Shell. In this case you'll see an entirely different screen:



MS-DOS Shell
File Options View Tree Help
[A:] [B:] [C:]
Directory Tree
C:\*.*
Main
F10=Actions Shift+F9=Command Prompt
2:08p

```
C:\
→[-] C:\
  ├─[ ] BAT
  ├─[ ] BIN
  ├─[ ] BTOOLS2
  ├─[ ] DOS
  ├─[+] GRAPHICS
  ├─[ ] PCLFONTS
  └─[ ] TEMP
```

```
▶ WINA20   .386        9,349  03-22-91
  AUTOEXEC.BAT           261  04-22-91
  COMMAND  .COM       47,845  03-22-91
  MIRROR   .FIL      122,368  04-19-91
  CONFIG   .SYS          286  04-24-91
  HIMEM    .SYS       11,304  05-01-90
  DEFAULT  .USP        6,112  04-04-91
```

```
Command Prompt
MS-DOS Editor
MS-DOS QBasic
[Disk Utilities]
```

The MS-DOS Shell

In the following section we'll work at both the command line level and with the MS-DOS Shell. Which method you choose depends on your personal preference. However, we'll show you how to switch between these two so that you don't always have to use the default setting selected during installation. We recommend that you work with both the command line and the MS-DOS Shell so you can determine which one you prefer.

From the command line to the MS-DOS Shell

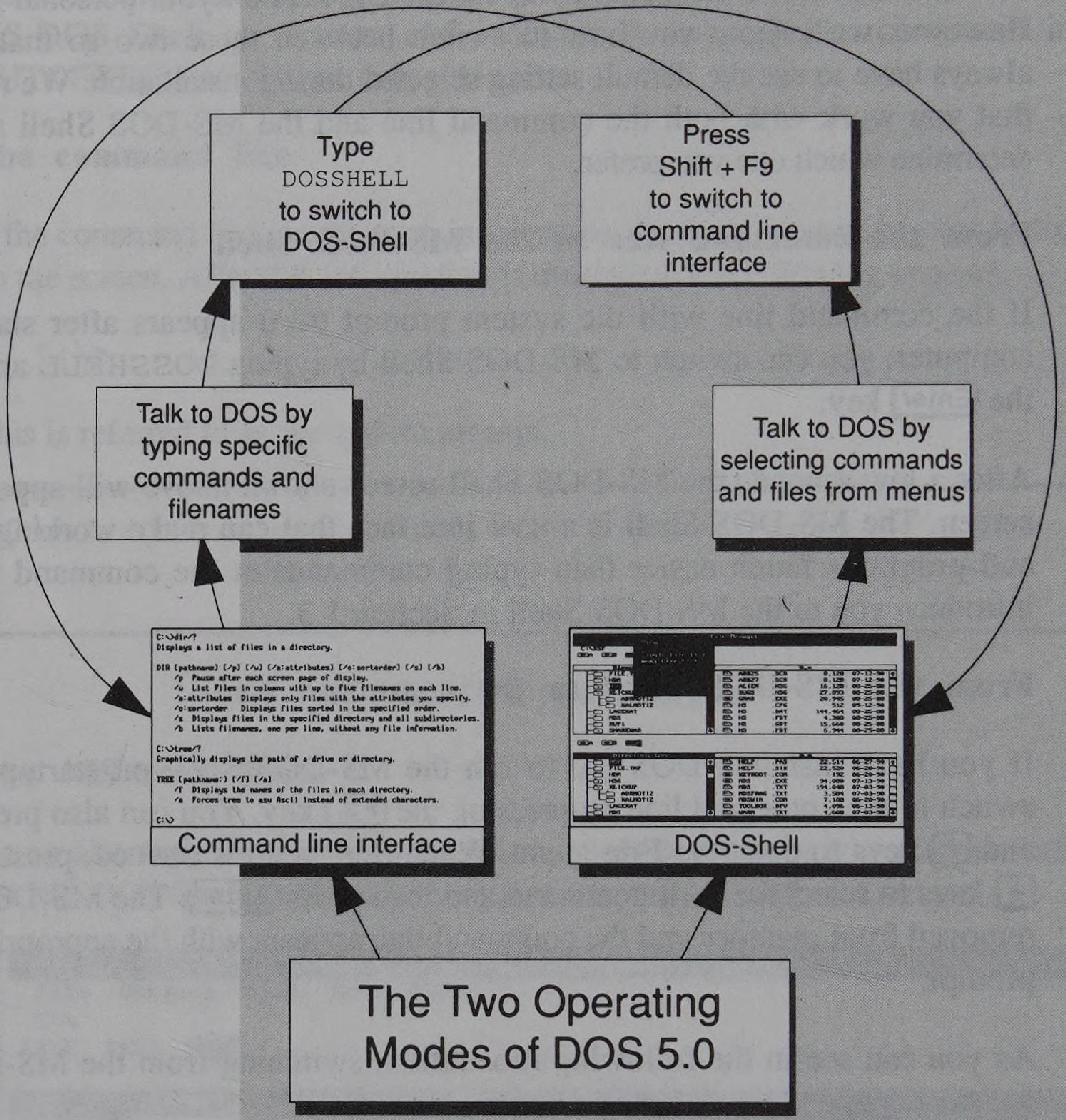
If the command line with the system prompt (C:\) appears after starting your computer, you can switch to MS-DOS Shell by typing `DOSSHELL` and pressing the Enter key.

After a few seconds the MS-DOS Shell screen shown above will appear on your screen. The MS-DOS Shell is a user interface that can make working with files and programs much easier than typing commands at the command line. We'll introduce you to the MS-DOS Shell in Section 1.3.

From the MS-DOS Shell to the command line

If you have installed DOS 5.0 to run the MS-DOS Shell on startup, you may switch to the command line by pressing the F3 key. You can also press the Alt and ↓ keys to open the **File** menu. When this menu is opened, press the ↑ or ↓ keys to select the **Exit** command and then press Enter. The MS-DOS Shell is removed from memory and the command line appears with the appropriate system prompt.

As you can see in the following illustration, switching from the MS-DOS Shell and the command line is easy and quick.


Type
DOSSHELL
to switch to
DOS-Shell
Press
Shift + F9
to switch to
command line
interface
Talk to DOS by
typing specific
commands and
filenames
Talk to DOS by
selecting commands
and files from menus
Command line interface
DOS-Shell
The Two Operating
Modes of DOS 5.0

```
C:\>dir/?
Displays a list of files in a directory.

DIR [pathname] [/p] [/w] [/a:attributes] [/o:sortorder] [/s] [/b]
    /p  Pause after each screen page of display.
    /w  List files in columns with up to five filenames on each line.
    /a:attributes  Displays only files with the attributes you specify.
    /o:sortorder   Displays files sorted in the specified order.
    /s  Displays files in the specified directory and all subdirectories.
    /b  Lists filenames, one per line, without any file information.

C:\>tree/?
Graphically displays the path of a drive or directory.

TREE [pathname:] [/f] [/a]
    /f  Displays the names of the files in each directory.
    /a  Forces tree to asu Ascii instead of extended characters.

C:\>
```

It's easy to switch between the command interface and the MS-DOS Shell

1.2 Using The Command Line

As previously mentioned, you can recognize the command line of MS-DOS by the system prompt:

```
C:\>
```

You can type MS-DOS commands at this prompt. Commands are programs that instruct your computer to perform certain tasks.

Next we'll use two commands to change the computer's date and time.

The DATE command

At the keyboard, type the following command and then press Enter.

```
DATE
```

After you press Enter, a message similar to the following appears on the screen:

```
Current Date is Wed 06-05-1991
Enter new date (mm-dd-yy):
```

If the date that the computer displays is different from the actual date, you should correct it by typing the actual date and pressing Enter. When doing this, remember to enter the date in the same form that the computer used in the bottom line. For example, if the date is July 4, 1991, then you would type the following line and press Enter:

```
07-04-91
```

It's not necessary to enter a zero if the day or month has only one digit. For example, if the date is July 3, 1991, you can simply type:

```
7-3-91
```

Pressing the Enter key returns the system prompt to the screen. This indicates that the command you entered has been processed and you can now type a new command.

The TIME command

This command displays the current system time. Type the following at the system prompt and press Enter:

```
TIME
```

After you press Enter, a message similar to the following, appears on the screen:

```
Current time is  4:15:52:06p
Enter new time:
```

This is your computer's current system time. It is so exact that it displays hundredth seconds. Compare the time on the screen with the actual time. Seconds and hundredth seconds are not necessary. If the time on the screen matches the actual time, simply press Enter to end the command.

When setting the correct time, remember to use the 24 hour clock that the computer displays on the screen. For time prior to noon, for example 10AM, just type 10:00 and press Enter. If the correct time is after noon (PM), add 12 to the hours. For example, 2PM is entered as 14:00, 3PM is 15:00 and so on. The most important thing to remember is that the colon must separate the hours from the minutes.

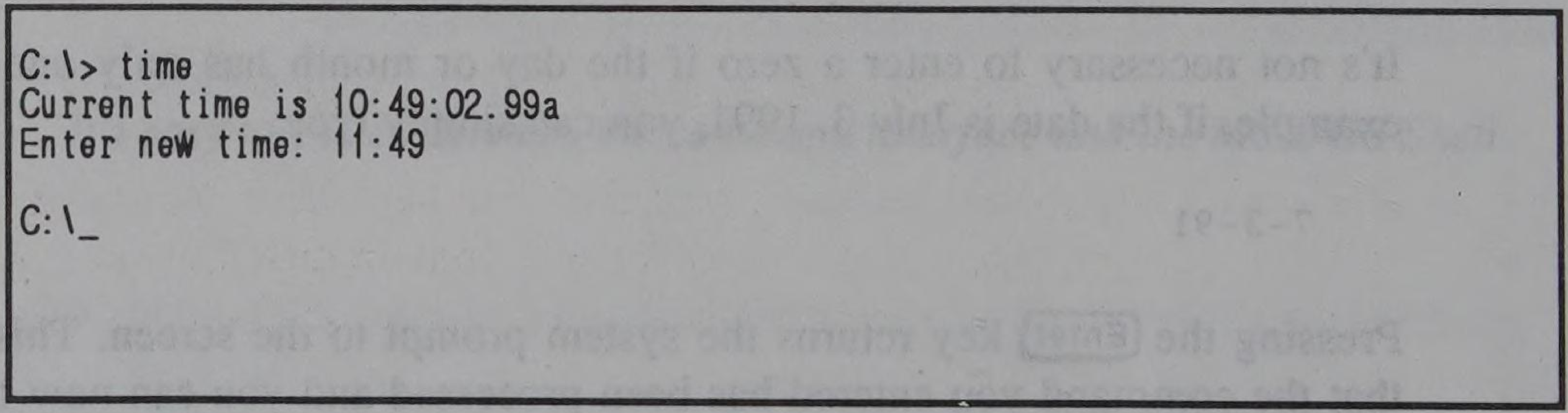
When you enter the time, simply type the hour and the minute. For example, if it's fifteen minutes after ten, then you would type the following:

```
10:15
```

It's not necessary to enter zeros if you're entering single digits. For example, if you type the following:

```
9:5
```

your computer will interpret this as five minutes after nine. When you press the Enter key, after typing the time, the system prompt returns to the screen. This indicates that the previous command was executed and you can enter a new command.



```
C:\> time
Current time is 10:49:02.99a
Enter new time: 11:49

C:\_
```

The Time command line

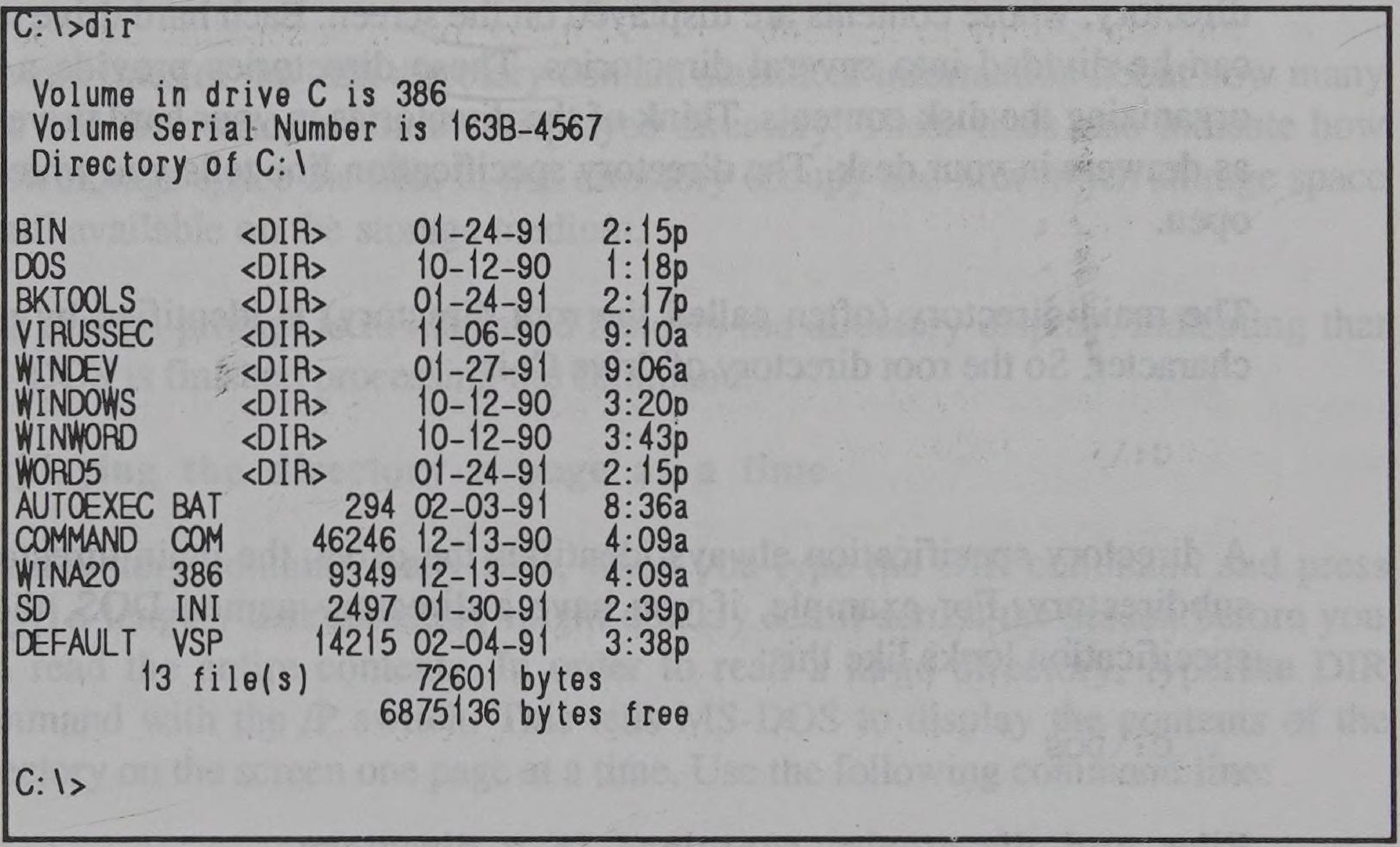
1.2.1 Displaying files

Now that we've used two simple MS-DOS commands, the DATE and TIME commands, we'll move on to a few others.

One of these commands instructs the computer to list the names of the files on your hard drive or floppy diskette. Simply type the following command after the system prompt:

```
DIR
```

Pressing the Enter key displays a screen similar to the following:



```
C:\>dir

 Volume in drive C is 386
 Volume Serial Number is 163B-4567
 Directory of C:\

BIN          <DIR>      01-24-91   2:15p
DOS          <DIR>      10-12-90   1:18p
BKTOOLS      <DIR>      01-24-91   2:17p
VIRUSSEC     <DIR>      11-06-90   9:10a
WINDEV       <DIR>      01-27-91   9:06a
WINDOWS      <DIR>      10-12-90   3:20p
WINWORD      <DIR>      10-12-90   3:43p
WORD5        <DIR>      01-24-91   2:15p
AUTOEXEC BAT       294 02-03-91   8:36a
COMMAND  COM     46246 12-13-90   4:09a
WINA20   386      9349 12-13-90   4:09a
SD       INI      2497 01-30-91   2:39p
DEFAULT  VSP     14215 02-04-91   3:38p
       13 file(s)      72601 bytes
                     6875136 bytes free

C:\>
```

A disk directory lists the contents of your hard drive

The screen display contains the contents of your hard drive. Your particular screen will be different since your hard drive will contain different files. A variety of useful information is displayed here. This *Table of Contents* of your disk is referred to as the *Disk Directory.*

Name of the storage medium

The disk drive designator (A:,B:,C:, etc.) and the volume label (if any) are displayed on the first line. If a volume label has not been assigned, the following message appears:

```
Volume in drive C has no label
```

Otherwise, the volume label, an electronic label on the diskette or hard drive, is displayed following the drive designator. You can see that the volume label above is 386.

The volume serial number

Every storage medium has a serial number which is used to identify it. MS-DOS often uses this serial number for control purposes. This number is displayed on the second line.

The directory specification

The next line contains the directory specification. This is the name of the directory, whose contents are displayed on the screen. Each hard drive and diskette can be divided into several directories. These directories provide a method of organizing the disk contents. Think of the directories in your hard drive or diskette as drawers in your desk. The directory specification line tells you which drawer is open.

The main directory (often called the root directory) is identified by a backslash character. So the root directory of drive C: is:

```
C:\
```

A directory specification always identifies the drive, the main directory and the subdirectory. For example, if you have a directory named DOS, the directory specification looks like this:

```
C:\DOS
```

Files and directories contained in a directory

The next lines display the files and subdirectories in this directory.

Subdirectories are identified by <DIR> (DIRECTORY) immediately following their names. All of the entries marked in this way are the next lower level of directories located within the current directory. The screen always displays the names of the subdirectories.

You can recognize files by the numeric values that follow the name. These values are the size of the file in bytes. It is important to remember that files usually consist of a filename and an extension. The file extension is important to MS-DOS and to you because it indicates the specific type of file.

Date and time

The column following the filename (and directories) contains the date and time the file or directory was created. You can use the date and time to determine which file was last worked on. So it's very important that the correct time and date are stored in your computer. Since most computers contain a battery operated clock for keeping the correct system time, they need only periodic updating.

When changes are made to files or directories, the date and time are changed and saved with the file. The new date and time are shown when you display the disk or directory contents with the DIR command.

The number of files and storage space

The last lines of the disk directory contain statistical information about how many files and directories are in the displayed directory. These lines also indicate how much storage space the files in this directory occupy and how much storage space is still available on the storage medium.

This system prompt is re-displayed beneath the directory display, indicating that MS-DOS is finished processing the command.

Displaying the directory a page at a time

If a directory contains many files, when you type the DIR command and press Enter a lengthy disk directory might quickly scroll across the screen before you can read the entire contents. In order to read a large directory, type the DIR command with the /P switch. This tells MS-DOS to display the contents of the directory on the screen one page at a time. Use the following command line:

```
DIR /P
```

The `P` represents PAGE. The /P switch is also called a parameter. Now, when you press Enter, the first lines of the disk directory won't scroll off the top of the screen. Instead, MS-DOS stops displaying the directory as soon as the screen is full. At the bottom of the screen, the following message appears:

```
Press any key to continue . . .
```

This message instructs you to press any key after you have viewed this section of the directory contents. Pressing any key will display the next screen (or page). When you reach the end of the directory, the system prompt appears on the screen.

Displaying the contents of a floppy diskette

To display the directory of a diskette, place the diskette in the drive and then make this disk drive the current drive. Use the companion diskette included with this book for the following examples. Insert the companion diskette into disk drive A:.

Changing current disk drives

Different disk drives are identified by disk drive letters. The first partition of a hard drive is identified with the letter C. Usually a disk drive is identified with the letter A. If you have two disk drives, the second one is called B. We will assume that you only have one disk drive, drive A:. In order to make this disk drive the current drive type, the disk drive identifier (A:) is followed by a colon at the system prompt, as the following shows:

```
A:
```

When you press Enter, the light on the disk drive will flash. If a correctly formatted diskette isn't in the disk drive, you will get an error message. If there is a correctly formatted floppy diskette in the disk drive, the system prompt changes to:

```
A:\>
```

The system prompt always indicates which disk drive is the current one.

Enter either DIR or DIR /P after the system prompt, as explained earlier. The screen will display the directory of the diskette. The information is formatted the same way as in our previous example.

To switch the current drive back to the C: partition of your hard drive, use the same procedure. Type the letter C, followed by a colon, after the system prompt and press Enter. The system prompt will then indicate that the current drive is the C: partition of your hard drive.

☞ If you have more than one partition on your hard drive, follow the same procedure for switching between partitions.

Changing directories

Important note: You can easily customize your system prompt. For example, you can choose to display the current time and date, change the text colors or print a special message as the prompt. The current disk drive and directory are displayed only if your prompt has been defined to include the characters PG. See Section 9.2, "Customizing The System Prompt", for more information on customizing the system prompt. The following discussion assumes that you haven't altered the default system prompt as established by the MS-DOS 5.0 installation.

As previously mentioned, there may be other subdirectories, within a directory, that contain their own files. When you display a directory, only the name of the subdirectory is displayed. How do you display the contents of these subdirectories? The companion diskette contains a directory name DOC. One method of displaying the contents of this directory is to make it the current directory. Make sure the companion diskette is in drive A:. Make sure A: is the current drive by typing:

```
A:
```

Type the following at the system prompt:

```
CD DOC
```

When you press Enter, the system prompt will change.

```
A:\DOC>
```

The system prompt not only displays the current drive, but also the current directory. The CD command represents Change Directory. This command allows you to change to another directory.

Once in this directory, type the command for displaying the contents of the directory.

```
DIR
```

The display is similar to the ones we've previously described.

If another subdirectory is contained within the current subdirectory, this subdirectory can also be changed to the current directory. For example, another directory called PRIVATE is located within the DOC subdirectory, it can be made the current directory by typing:

```
CD PRIVATE
```

After pressing Enter, the system prompt changes again.

```
A:\DOC\PRIVATE>
```

The system prompt indicates in which directory you're currently located. This directory (PRIVATE) is a subdirectory of DOC, which the system prompt clearly indicates. This is also called the *path specification*.

The contents of the PRIVATE directory can also be displayed on the screen using the DIR command.

To display the contents of the DOC directory again, you must move back a level in the directory level. This can no longer be done by entering the name of the directory after the CD command. This command is used to branch one level deeper in the directory structure. To return to a higher level in the directory structure, type CD and two periods, as in the following command:

```
CD ..
```

After pressing Enter, the system prompt will indicate that the current directory is the DOC directory. The two periods after the CD command always designate the next highest directory of MS-DOS.

You can return to the main or *root* directory of the current drive by typing the following:

```
CD \
```

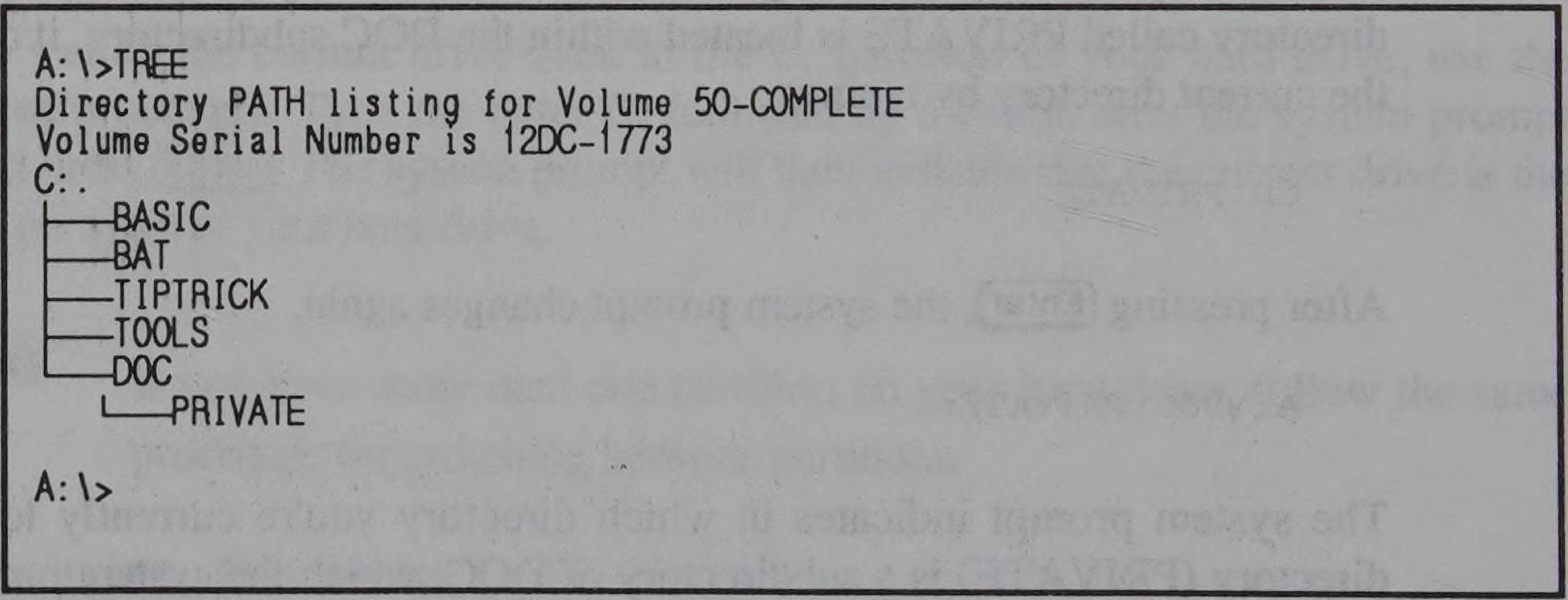
In this example, when you press Enter, you will see the same system prompt that you saw at the beginning.

```
A:\>
```

Using the CD command, followed by the backslash character (CD \), causes MS-DOS to bypass all higher subdirectories and move directly to the root directory of the current disk drive.

If the system prompt consists of only the drive letter and the backslash without any subdirectory names, you're located in the main directory of the current drive.

The TREE command will display the directory structure of a disk.



```
A:\>TREE
Directory PATH listing for Volume 50-COMPLETE
Volume Serial Number is 12DC-1773
C:.
├───BASIC
├───BAT
├───TIPTRICK
├───TOOLS
└───DOC
    └───PRIVATE

A:\>
```

Directory Tree structure

1.2.2 Copying files

The COPY command is another important MS-DOS command. It is used to copy files from the hard drive to a floppy diskette or vice versa.

There are different reasons for copying files. For example, you can copy files that you've created on your computer, to a diskette, either as a backup copy of the file or to give to someone else, who can copy the file from the diskette to their computer's hard drive.

Copying from the hard drive to a diskette

Use this procedure to copy a file from your hard drive to a floppy diskette. First, you must know the name of the file. This can be determined by using the DIR command. It's important that you know the entire filename exactly as it was recorded in the disk directory. For example, you must know the name itself and the filename extension.

Suppose that you want to copy the AUTOEXEC.BAT file, which is in the root directory of your hard drive, onto a diskette. You already know the name of the file. Enter this name after the COPY command and tell MS-DOS to which location you want to copy the file.

In this example, we'll copy AUTOEXEC.BAT to a diskette which is in disk drive A:. Type this disk drive identifier, followed by a colon. The command is as follows:

```
COPY C:\AUTOEXEC.BAT A:
```

After you press Enter, the disk drive light will be on, indicating that the disk drive is busy. After a short time, the following message from DOS appears:

```
1 file(s) copied
```

This tells you that the file has been copied to the target disk. Another indication that the file transfer is completed is that the system prompt will reappear. Make sure A: is the current drive by typing:

```
A:
```

The DIR command can be used to display the directory contents of the diskette, which will now contain a copy of the AUTOEXEC.BAT file. Type the following to change the current drive back to the hard drive:

```
C:
```

You've just copied a file from the hard drive onto the diskette. But what about copying a file, which is stored in the DOS directory? This is just as easy to do. Make the DOS directory the current directory by typing:

```
CD DOS
```

After pressing Enter, the system prompt will change to "C:\DOS", indicating the current directory. The COPY command now works the same way as described above. Enter COPY, followed by the name of a file, for example FORMAT.COM, and the target drive, in this case, drive A:. The command line looks like the following:

```
COPY FORMAT.COM A:
```

When copying, remember to make the directory, which contains the file to be copied, the current directory. This makes using the COPY command very easy. After this command, enter the name of the file, including the file extension. Then enter the target disk drive, followed by a colon.

If the file isn't located in the current directory or if the name of the file was entered incorrectly, MS-DOS will display an error message, which informs you of the error.

```
File not found - FIRMAT.COM
            0 file(s) copied
```

Copying from a diskette to the hard drive

Copying a file from a diskette onto your hard drive works the same way. Use the following procedure to copy the AUTOEXEC.BAT file, which you copied to your floppy diskette, onto your hard drive. First make sure that the root directory of the hard drive is the current drive by typing the following:

```
CD \
```

The system prompt should look similar to the following:

```
C:\>
```

Now change the current drive to A: by typing:

```
A:
```

Use the COPY command and type the name of the file and the target drive after this command. In this example, C: is the target drive. The command line looks like the following:

```
COPY AUTOEXEC.BAT C:
```

MS-DOS will display a message informing you that it has copied a file. The system prompt will then reappear.

Remember that when you do this, the AUTOEXEC.BAT file on your hard drive is overwritten by the AUTOEXEC.BAT file that was copied from your diskette. Don't change this file until you understand the effects of the commands contained in the AUTOEXEC.BAT file of your boot procedure.

How do you copy the LETTER.DOC file from the PRIVATE directory on the companion diskette to the hard drive? Use the following procedure to make the PRIVATE directory on the companion the current directory. Set the current drive to drive A by typing:

```
A:
```

Then change the current directory to the PRIVATE directory by first changing to the DOC directory.

```
CD DOC
```

Next, enter the following to make PRIVATE the current directory:

```
CD PRIVATE
```

Again, the system prompt will inform you that this is the current directory. Change drives by typing:

```
C:
```

At this point, you can use the COPY command to copy the file. The command line looks like the following:

```
COPY A:LETTER.DOC C:
```

When copying from the diskette onto the hard drive, remember to make the directory, to which you want to copy the file (the target directory), the current drive on the hard drive. It's also possible to copy files by specifying the complete path; this method is generarally much easier and less prone to errors.

1.2.3 Looking at files

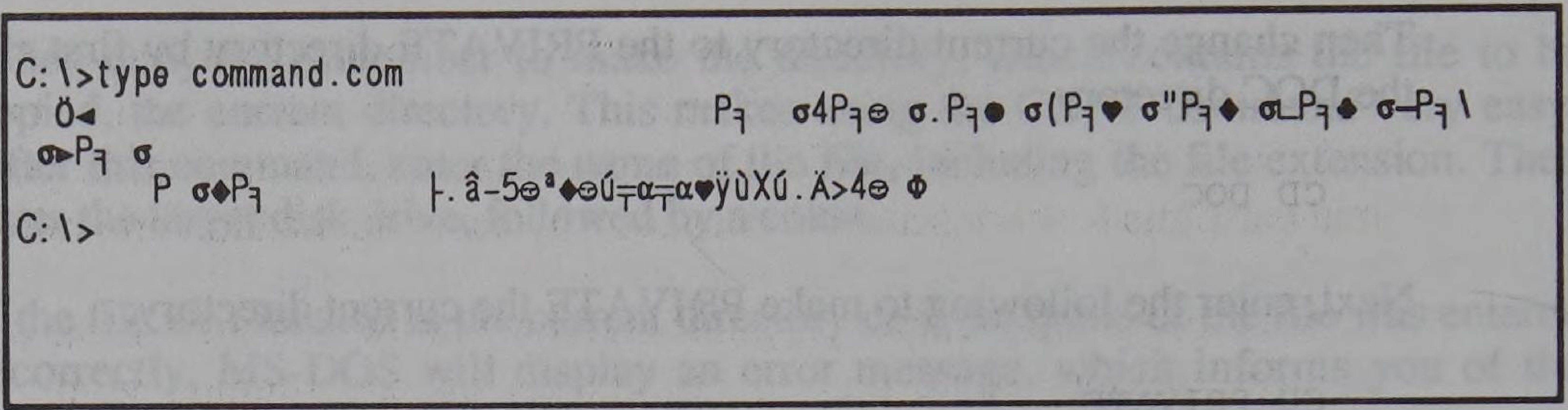
So far we've displayed the contents of a directory and copied files. Now, let's work with files and their contents.

It's not easy to display the contents of files. Although MS-DOS contains the TYPE command, which displays the contents of files on the screen, this command is only suitable for displaying ASCII or pure text files.

Experiment with the TYPE command. Enter TYPE at the command line after the system prompt, followed by the name of the file to be displayed. Suppose that you want to display the contents of the COMMAND.COM file from the root directory of your hard drive. Make the root directory the current directory and enter the following command line:

```
TYPE COMMAND.COM
```

After you press Enter, your computer will beep and display some strange characters on the screen. You won't be able to identify many of these characters.



```
C:\>type command.com
Ö◄                                 P╕  σ4P╕⊖ σ.P╕● σ(P╕♥ σ"P╕♦ σ┴P╕♣ σ┬P╕\
σ►P╕ σ
     P σ♦P╕             ├.â-5⊖ª♦⊖ú╤α╤α♥ÿùXú.Ä>4⊖ Φ
C:\>
```

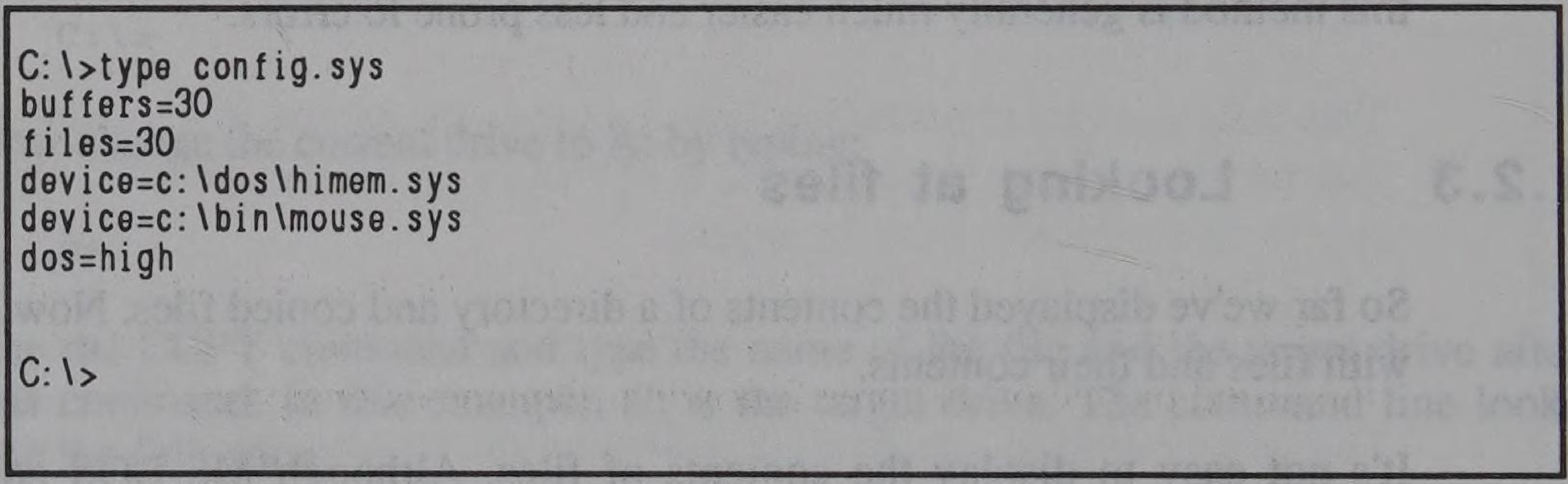
Using the TYPE command to display the COMMAND.COM file

These strange characters appear because COMMAND.COM is a program file. Remember that only ASCII or pure text files can be displayed with the TYPE command.

To view a text file, type the following:

```
TYPE CONFIG.SYS
```

The contents of a typical CONFIG.SYS file are displayed on the screen. This file is a pure text file, containing instructions to the operating system for operations conducted during the startup.



```
C:\>type config.sys
buffers=30
files=30
device=c:\dos\himem.sys
device=c:\bin\mouse.sys
dos=high

C:\>
```

Using the TYPE command to display the CONFIG.SYS file

You'll soon discover that most files are linked to certain application programs and can only be viewed correctly from within that application. The illustration on page 18 should clarify this for you.

1.2.4 Starting and quitting programs

To display the contents of different files, it's best to start the application program that's associated with the file. So, to view a file created with a database, run the database program that created it.

With these application programs you can view existing files and create new ones. Application programs are the main reason for using any computer. After all, you purchased your computer because of its ability to handle a database, run a word processor or spreadsheet, etc. These are all things which cannot be done from MS-DOS. However, MS-DOS (or some form of DOS) is vital to your computer and performs functions and services used by many application programs.

If you have an application program, but don't know how to start it, the following section provides some tips on how to get up and running with virtually any program.

Starting programs

Application programs almost always store their associated files in their own directories. Change the current directory to the directory in which these files are located.

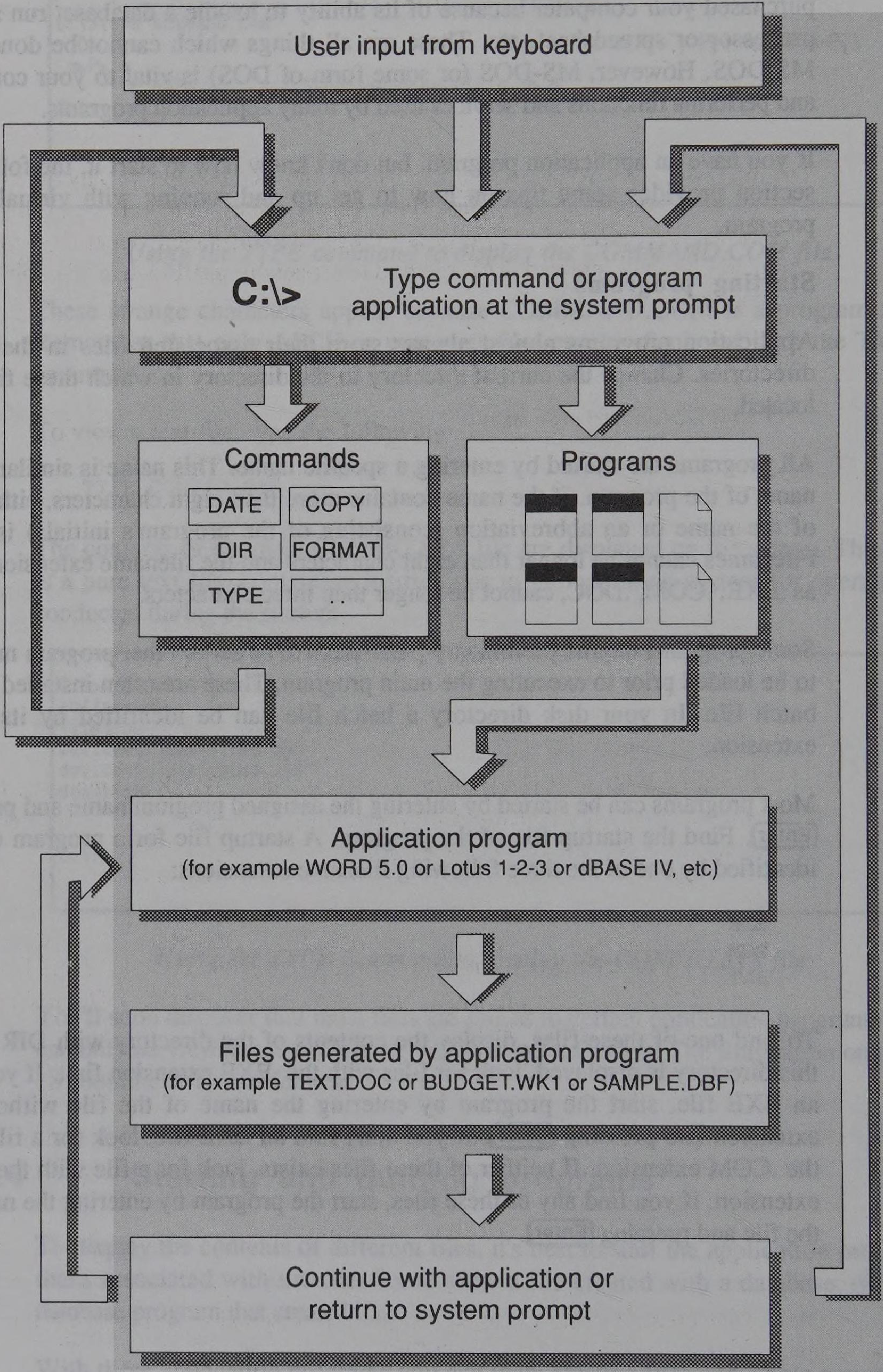
All programs are started by entering a specific name. This name is similar to the name of the program. If the name contains more than eight characters, either part of the name or an abbreviation (consisting of the program's initials) is used. Filenames cannot be longer than eight characters and the filename extension, such as .EXE, .COM, .DOC, cannot be longer than three characters.

Some programs require preliminary parameters to be set or other program modules to be loaded prior to executing the main program. These are often installed with a batch file. In your disk directory a batch file can be identified by its .BAT extension.

Most programs can be started by entering the assigned program name and pressing Enter. Find the startup file of the program. A startup file for a program can be identified by one of the three following filename extensions:

```
EXE
COM
BAT
```

To find one of these files, display the contents of the directory with DIR. Once this directory is displayed, look for files with the .EXE extension first. If you find an EXE file, start the program by entering the name of the file without the extension and pressing Enter. If you don't find an EXE file, look for a file with the .COM extension. If neither of these files exists, look for a file with the .BAT extension. If you find any of these files, start the program by entering the name of the file and pressing Enter.


User input from keyboard
C:\>
Type command or program application at the system prompt
Commands
DATE
COPY
DIR
FORMAT
TYPE
. .
Programs
Application program
(for example WORD 5.0 or Lotus 1-2-3 or dBASE IV, etc)
Files generated by application program
(for example TEXT.DOC or BUDGET.WK1 or SAMPLE.DBF)
Continue with application or
return to system prompt

Files are linked to specific application programs

You can also apply a *filter* to the DIR command by typing one of the following:

```
DIR *.EXE
DIR *.COM
DIR *.BAT
```

The asterisk (*) is a wildcard character. This character lets you display the names of all files that end with the specified filename extension. In the above examples, all files which end with EXE, COM or BAT would be displayed, depending on which command line was entered. This method is especially useful when searching large directories.

The following are a few examples of this method:

Microsoft Word word processor

The directory of the word processing program Microsoft Word contains a file named WORD.EXE. Since this file has the same name as the application program, it's probably the startup file. To make sure, enter the following and press Enter:

```
WORD
```

Most files created using Microsoft Word have the extension .DOC. This extension represents a document file.

WordPerfect word processor

The directory of WordPerfect contains a file named WP.EXE. Since this file has the same name as the application program, it's probably the startup file. To make sure, enter the following and press Enter:

```
WP
```

There isn't a standard extension with WordPerfect when creating files. However, you can set one up in the program. You may want to use the .DOC extension to indicate that it is a document file.

Lotus 1-2-3 spreadsheet

The directory of Lotus 1-2-3 contains a file named LOTUS.EXE. Since this file has the same name as the application program, it's probably the startup file. To make sure, enter the following and press Enter:

```
LOTUS
```

Most files created using Lotus 1-2-3 have the extension .WKS or .WK1. These extensions represent the spreadsheet files.

dBASE IV data management system

The directory of dBASE IV contains a file named DBASE.EXE. Since this file has the same name as the application program, it's probably the startup file. To make sure, enter the following and press Enter:

```
DBASE
```

Most files created using dBASE IV have the extension .NDX or .DBF. These extensions represent index files and database files.

WINDOWS

In the WINDOWS directory you'll find many files with the .EXE file extension but none of them are true startup files. However, there is a file named WIN.COM. Enter the following and press Enter:

```
WIN
```

Exiting application programs

When you want to quit using an application program you should return to the MS-DOS command line. However, simply switching off the computer could cause you to lose important data.

If you're not sure how to exit a program and the documentation isn't available, the following are a few methods used by many programs. Most application programs use menus which contain different functions. These menus usually contain a function for exiting the application program, saving all files, closing windows, etc.

These menus are usually accessed by pressing Esc or Alt. Use the cursor keys to choose the different menu functions, then press Enter to activate them. Some application programs use a special menu for exiting the program. After selecting the exit option, you will return to the MS-DOS command line.

The following commands are used for exiting most application programs:

QUIT, EXIT, END, or something similar.

Choose the appropriate command and press Enter. In many programs a security prompt, which asks you to confirm whether or not you really want to leave the program, appears.

The MS-DOS Shell

To exit the MS-DOS Shell, press Alt F, then select the **Exit** command and press Enter. You'll be returned to the MS-DOS command line.

WINDOWS

Press Alt and then F to open the File menu, then select **Exit Windows** and press Enter. A security prompt, asking whether or not you really want to end the program, will appear. Press Enter to return to the MS-DOS command line.

1.2.5 Deleting files

To remove an unnecessary file from a hard drive or a diskette, use the DEL command.

To delete a specific file, you must know the exact name of the file and its extension. Enter the complete filename after the DEL command.

Be very careful when deleting files because once a file is deleted, it will be permanently lost. Although there are some utility programs for undeleting files, including an UNDELETE command from MS-DOS 5.0, it may be impossible to recover a deleted file. So before deleting a file make sure that you no longer have any use for it.

As an example, let's delete the LETTER.DOC file from your diskette in drive C:. We copied this file from the companion diskette to the hard drive in an earlier example. First change the drive, in which the diskette is located, to the current drive. Type the following:

```
C:
```

After you press Enter, the disk drive will be the current drive, as indicated by the system prompt. Enter the DEL command to erase the file and enter the name of the file you want to delete (in this case, LETTER.DOC). The command line should look like the following:

```
DEL LETTER.DOC
```

When deleting files, it's important to make the drive and directory, in which the file is to be deleted, the current drive and directory. This prevents you from deleting a file, which has the same name, in the wrong drive or different directory.

Use the DIR command to check the current directory before deleting any files. Checking the date on the file will also indicate whether the file is old or has been recently created.

1.2.6 Formatting diskettes

When you buy a new box of diskettes, you must specially prepare them before you can write information on them. Preparing a diskette for use is called "formatting the disk". To do this, use the FORMAT command. If you try to write to an unformatted diskette, MS-DOS will return an error message.

To format a disk, place the new diskette into drive A: and type the following command:

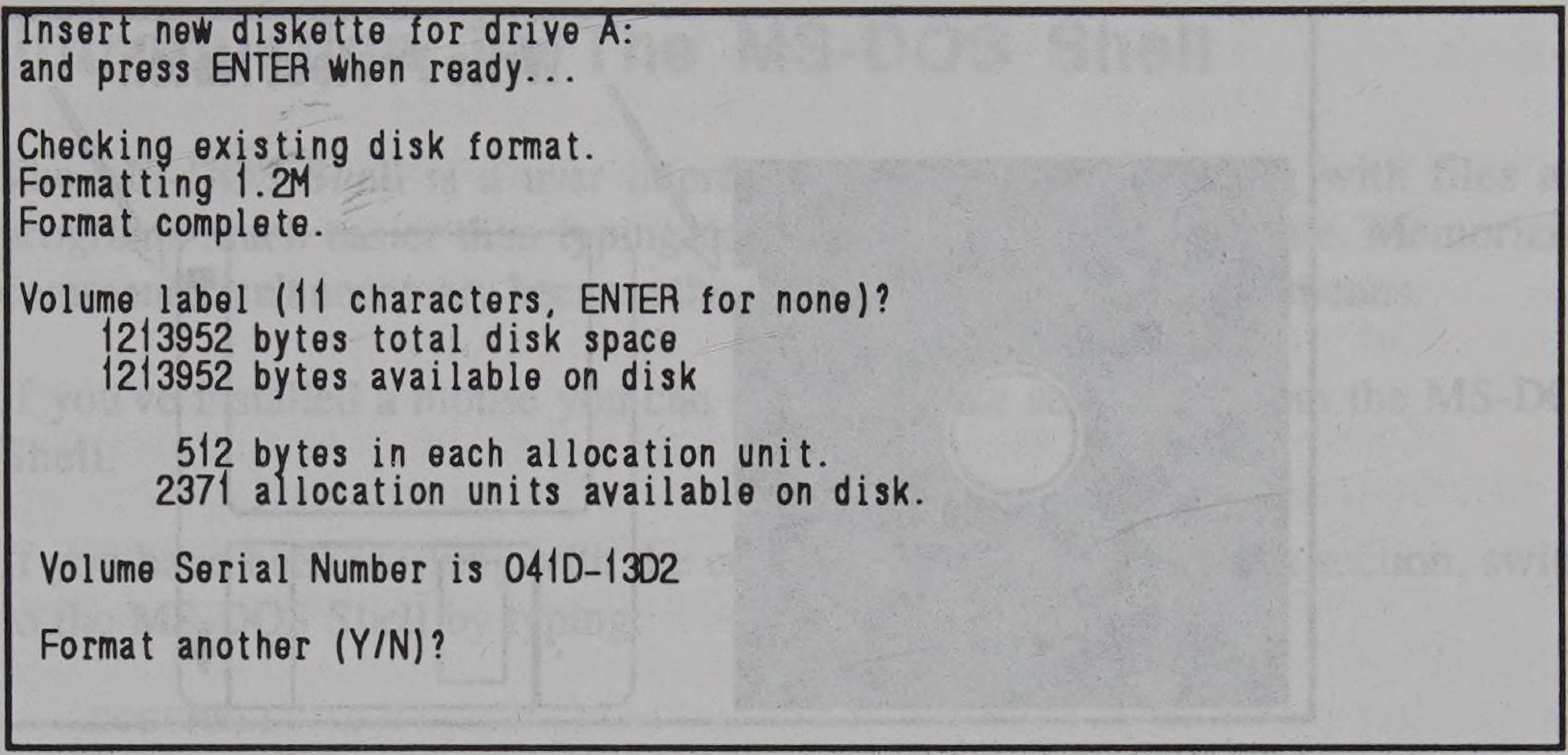
```
FORMAT A:
```

You're telling your computer that it should prepare the storage medium (diskette) in drive A: to store data. After pressing Enter, MS-DOS will prompt you to place a diskette in drive A: and press Enter. If the diskette to be formatted is already in drive A:, press Enter. The computer will begin formatting the diskette. It takes some time for the computer to format the diskette. The length of time depends on several factors, including the type of computer and disk drive.

After the formatting process is finished, MS-DOS prompts you to enter a volume name (also called a volume label). This is the unique name assigned to the diskette. It's displayed on the first line of the directory. You may enter a name up to eleven characters long and press Enter. If you prefer not to enter a volume name for the diskette, simply press Enter.

Then several lines of information, indicating how much storage space is available on the diskette, are displayed.

The next message asks whether you would like to format another diskette. Press Y to format another diskette or N to cancel the format command.



```
Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format.
Formatting 1.2M
Format complete.

Volume label (11 characters, ENTER for none)?
    1213952 bytes total disk space
    1213952 bytes available on disk

        512 bytes in each allocation unit.
       2371 allocation units available on disk.

 Volume Serial Number is 041D-13D2

 Format another (Y/N)?
```

Be careful when formatting diskettes. If you format a used diskette, it probably already contains data. All of the previous data is lost when you format the diskette. Before formatting a diskette, check whether you've inserted the correct one. Also check the directory of a previously formatted disk before formatting it. Type the command line:

```
DIR A:
```

If an error message is displayed on the screen, you can be relatively certain that you have an unformatted diskette, which must be formatted before it can be used. To cancel the error message, press A (for "Abort").

The system prompt returns to the screen and you can enter the format command.

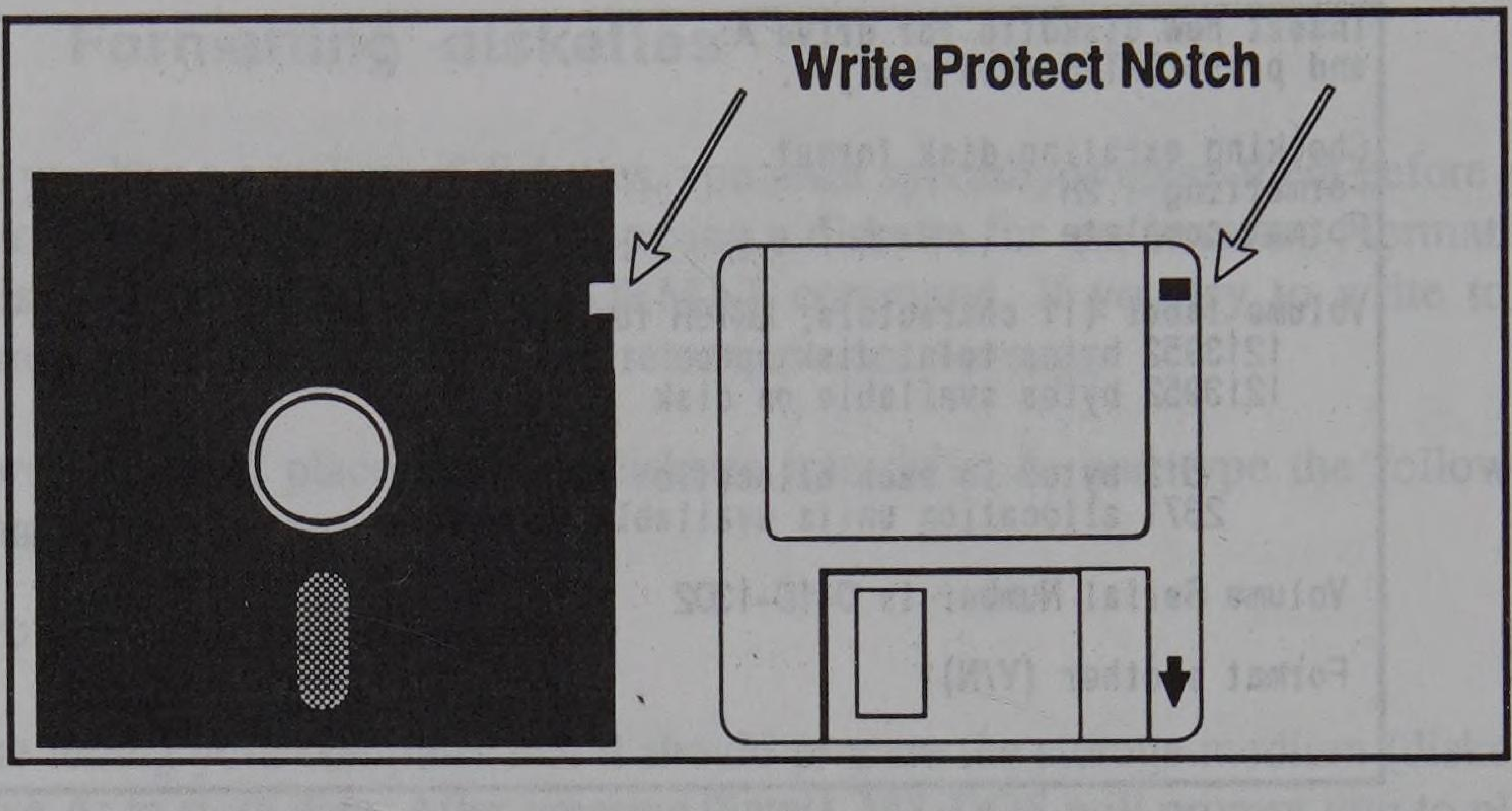
1.2.7 Copying diskettes

Occasionally you'll want to copy an entire diskette. For example, you might want to create backup copies of your program diskettes or you might want to send someone a disk containing all of the files on a particular diskette.

When you use the DISKCOPY command to copy a diskette, the entire diskette will be copied. The parts that contain data as well as the places on the diskette that are blank are copied. The DISKCOPY command works like a copying machine.

Before copying diskettes, you should safeguard the original diskette by write protecting it.

- For a 5 1/4" diskette, cover the write protect notch with an adhesive tab.
- For a 3 1/2" diskette, slide the small tab so that the hole is exposed.



Locations of the write protect notch on 5 1/4" and 3 1/2" diskettes

To copy a diskette, you will need a blank diskette. Type the following command line:

```
DISKCOPY A: A:
```

After pressing Enter, you're prompted to place the source diskette (the original diskette you want to copy) into drive A: and press any key. In a few seconds you are prompted to place the target diskette into drive A:. Insert the blank diskette and press any key.

If you're working with diskettes that have a storage capacity of 1.2 Meg, the source diskette and the target diskette must be swapped several times. The screen will display a message that indicates when to do this.

After the entire diskette has been copied, you are asked if you want to copy another diskette. Press Y if you want to do this. You will be prompted to insert another diskette; when this is done the entire process is repeated.

If you don't want to copy more diskettes, press N. The DISKCOPY command is now finished and the system prompt is displayed.

Be careful when copying diskettes. If you insert a diskette, which contains data, as your target diskette, the data will be overwritten without any warning.

1.3 Introduction To The MS-DOS Shell

The MS-DOS Shell is a user interface that can make working with files and programs much easier than typing commands at the command line. Memorizing commands isn't necessary because they can be controlled from the menus.

If you've installed a mouse you can use it to make selections from the MS-DOS Shell.

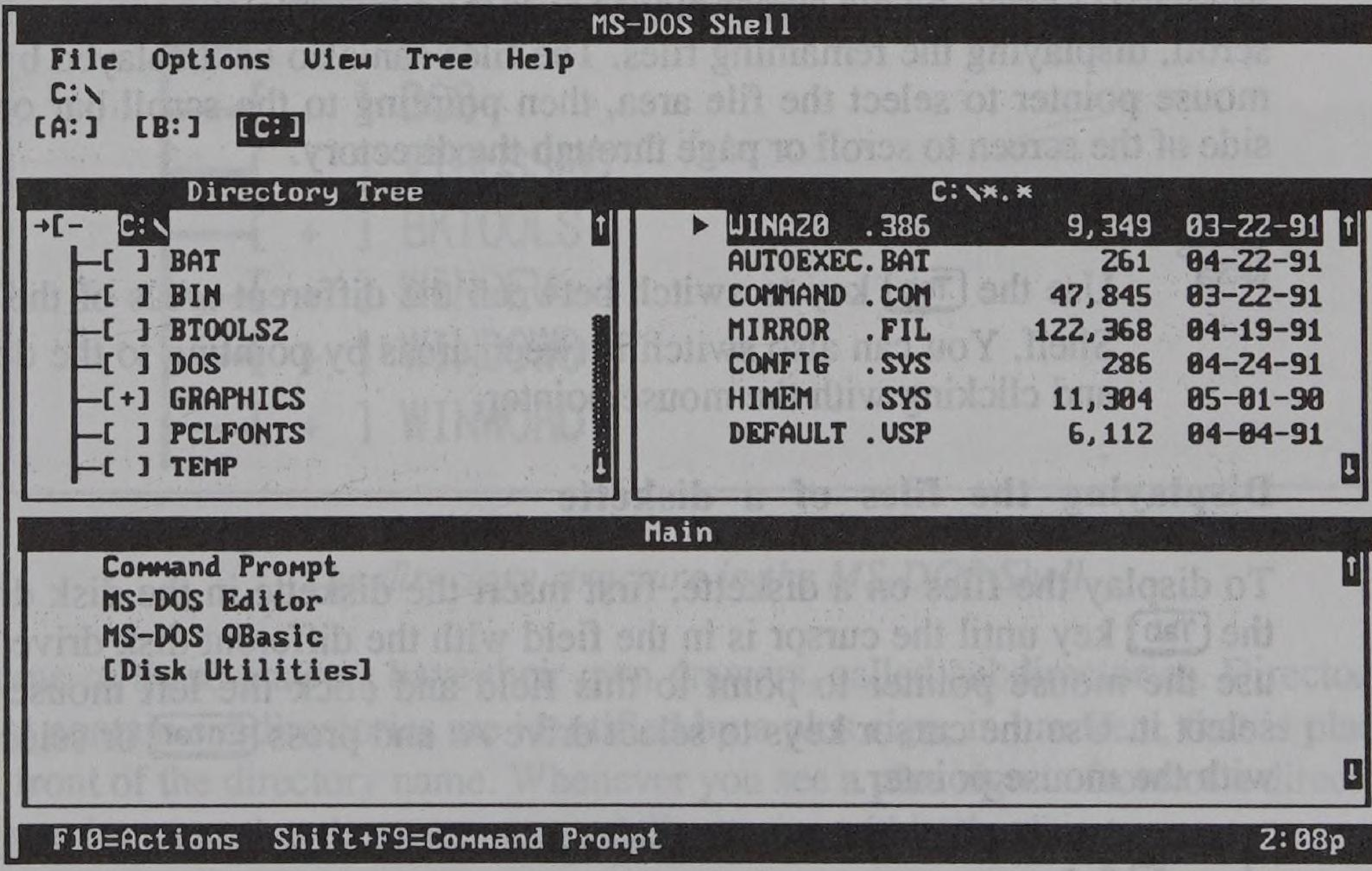
If you have been working with the command line in the previous section, switch to the MS-DOS Shell by typing:

```
DOSSHELL
```

The MS-DOS Shell screen appears.

1.3.1 Displaying files with the MS-DOS Shell

The MS-DOS Shell always displays the files in the current directory on the right center half of the screen. The file information displayed by the MS-DOS Shell looks like the following:



```
                              MS-DOS Shell
 File  Options  View  Tree  Help
 C:\
[A:]  [B:]  [C:]
          Directory Tree                          C:\*.*
→[-  C:\                          ↑   ▶ WINA20  .386      9,349  03-22-91 ↑
   ├─[ ] BAT                              AUTOEXEC.BAT        261  04-22-91
   ├─[ ] BIN                              COMMAND .COM     47,845  03-22-91
   ├─[ ] BTOOLS2                          MIRROR  .FIL    122,368  04-19-91
   ├─[ ] DOS                              CONFIG  .SYS        286  04-24-91
   ├─[+] GRAPHICS                         HIMEM   .SYS     11,304  05-01-90
   ├─[ ] PCLFONTS                         DEFAULT .VSP      6,112  04-04-91
   └─[ ] TEMP                     ↓                                         ↓
                                     Main
      Command Prompt                                                        ↑
      MS-DOS Editor
      MS-DOS QBasic
      [Disk Utilities]
                                                                            ↓
 F10=Actions  Shift+F9=Command Prompt                                 2:08p
```

The files in the MS-DOS Shell

☞ The MS-DOS Shell operates in graphic mode and text mode. For clarity we will use text mode in the following explanations. Your MS-DOS Shell screens will appear slightly different if you are in graphics mode.

You may wish to switch to the text mode to run the following examples. Press the Alt key then select the **Options** menu (by pressing the O key) and the **Display** command (by pressing the D key), then select a text mode.

The file sign

If your computer contains a graphic card and you're working in graphics mode, a sign or symbol that represents the type of file you are dealing with appears. The MS-DOS Shell differentiates files according to their use. For example, a different symbol is used for files that you created with a word processor than for other types of files.

The filename

The filename and its extension follows the symbol. The extension is separated from the name by a period. This file extension is significant to MS-DOS because it can recognize the type of file by its extension.

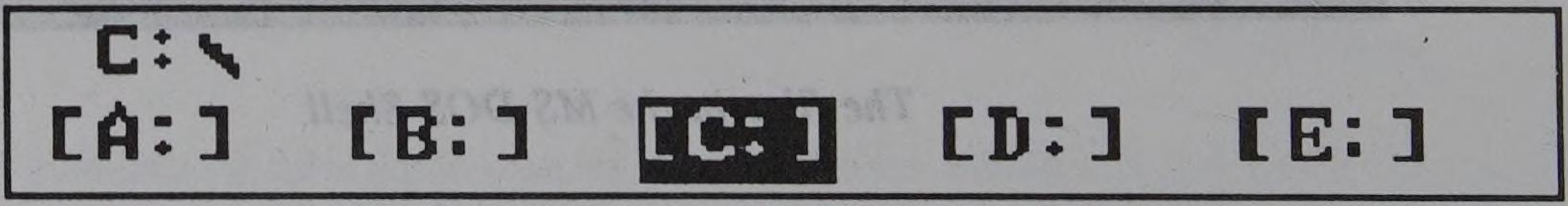
The size of the file appears next and then the date the file was created or last updated is displayed. This date indicates when you last used the file.

If there are more files in the directory than can fit on the screen, the rest of the files can be displayed by pressing the Tab key until the file display area is highlighted. Press the ↓ key. As the cursor moves over the filenames, the screen will begin to scroll, displaying the remaining files. The files can also be displayed by using the mouse pointer to select the file area, then pointing to the scroll bar on the right side of the screen to scroll or page through the directory.

☞ Use the Tab key to switch between the different areas of the MS-DOS Shell. You can also switch between areas by pointing to the desired area and clicking with the mouse pointer.

Displaying the files of a diskette

To display the files on a diskette, first insert the diskette in the disk drive. Press the Tab key until the cursor is in the field with the different disk drive labels. Or use the mouse pointer to point to this field and click the left mouse button to select it. Use the cursor keys to select drive A: and press Enter or select drive A: with the mouse pointer.



```
C:\
[A:]  [B:]  [C:]  [D:]  [E:]
```

The drive icons in text mode of the MS-DOS Shell

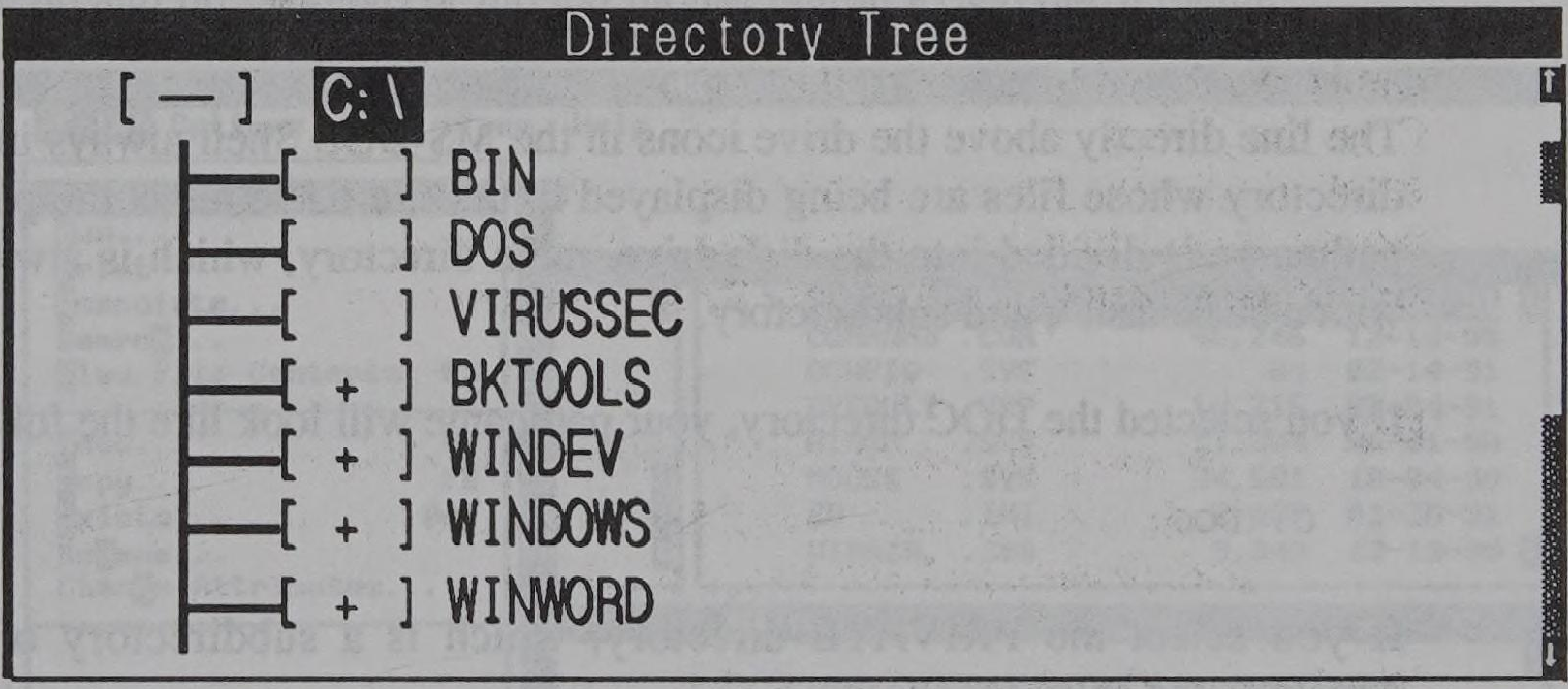
After a few seconds the MS-DOS Shell displays the files contained on the diskette.

To return to the hard drive, use Tab to select the drive icons and move the cursor to drive letter C (always the name of the hard drive) and press Enter or select drive C: with the mouse pointer.

Changing directories

On the left half of the screen the MS-DOS Shell displays the directory structure of the selected drive (in this case, C:). Remember that a hard drive or diskette can be divided into several directories, which provide an overview of the hard drive or diskette contents. Think of the hard drive or diskette as a desk with different drawers, called directories. In this directory structure you can immediately see which "drawers" are available. The name of the directory is always displayed.

To look at the contents of a different directory, use Tab to move the cursor to the field with the directory structure. Otherwise point to it with the mouse and click the left mouse button to select it. Then use the cursor keys to select the directory you want to see. The right half of the screen always displays the files in the selected directory.



```
Directory Tree
[—] C:\
   ├──[  ] BIN
   ├──[  ] DOS
   ├──[  ] VIRUSSEC
   ├──[+] BKTOOLS
   ├──[+] WINDEV
   ├──[+] WINDOWS
   └──[+] WINWORD
```

The directory structure in the MS-DOS Shell

Some of these drawers have their own drawers, called subdirectories. Directories that contain subdirectories are identified by a plus sign, in brackets, that is placed in front of the directory name. Whenever you see a plus sign in front of a directory name, it means that there are more subdirectories within the directory.

To display these subdirectories along with their contents, move the cursor to a directory that has a plus sign and press the + key. This directory opens and the names of the subdirectories (indented) are displayed. You can use the cursor keys or the mouse pointer to select these directories. The file list area displays the names of the files contained in the selected directory.

Suppose that you have a directory on your hard drive named DOC, which has a subdirectory called PRIVATE. At first, only the DOC directory would be displayed in the directory structure. This directory is marked with a plus sign, similar to the following:

```
|-[+] DOC
```

Use the cursor keys or the mouse to select this directory and press the + key or click with the left mouse button. Now the MS-DOS Shell will also display the PRIVATE directory. The entry in the directory list changes and looks like the following:

```
|-[-] DOC
   |-[ ] PRIVATE
```

It's also possible to "close" these directories so that they are no longer displayed on the screen. To do this, move the cursor to the directory, which is now marked with a minus sign, and press the - key or click the left mouse button. The subdirectories disappear and the minus sign changes to a plus sign. This indicates that the subdirectories are available but aren't being displayed.

Displaying the name of a directory

The line directly above the drive icons in the MS-DOS Shell always indicates the directory whose files are being displayed in the file list. This is the pathname. A pathname is divided into the disk drive, main directory, which is always marked with a backslash \, and subdirectory.

If you selected the DOC directory, your pathname will look like the following:

```
C:\DOC
```

If you select the PRIVATE directory, which is a subdirectory of the DOC directory, the entry changes to:

```
C:\DOC\PRIVATE
```

1.3.2 Copying files with the MS-DOS Shell

There are various reasons for copying files. For example, you might want to copy files that you created on your computer onto a diskette so that you have a backup copy of them.

Copying files from the hard drive to a diskette

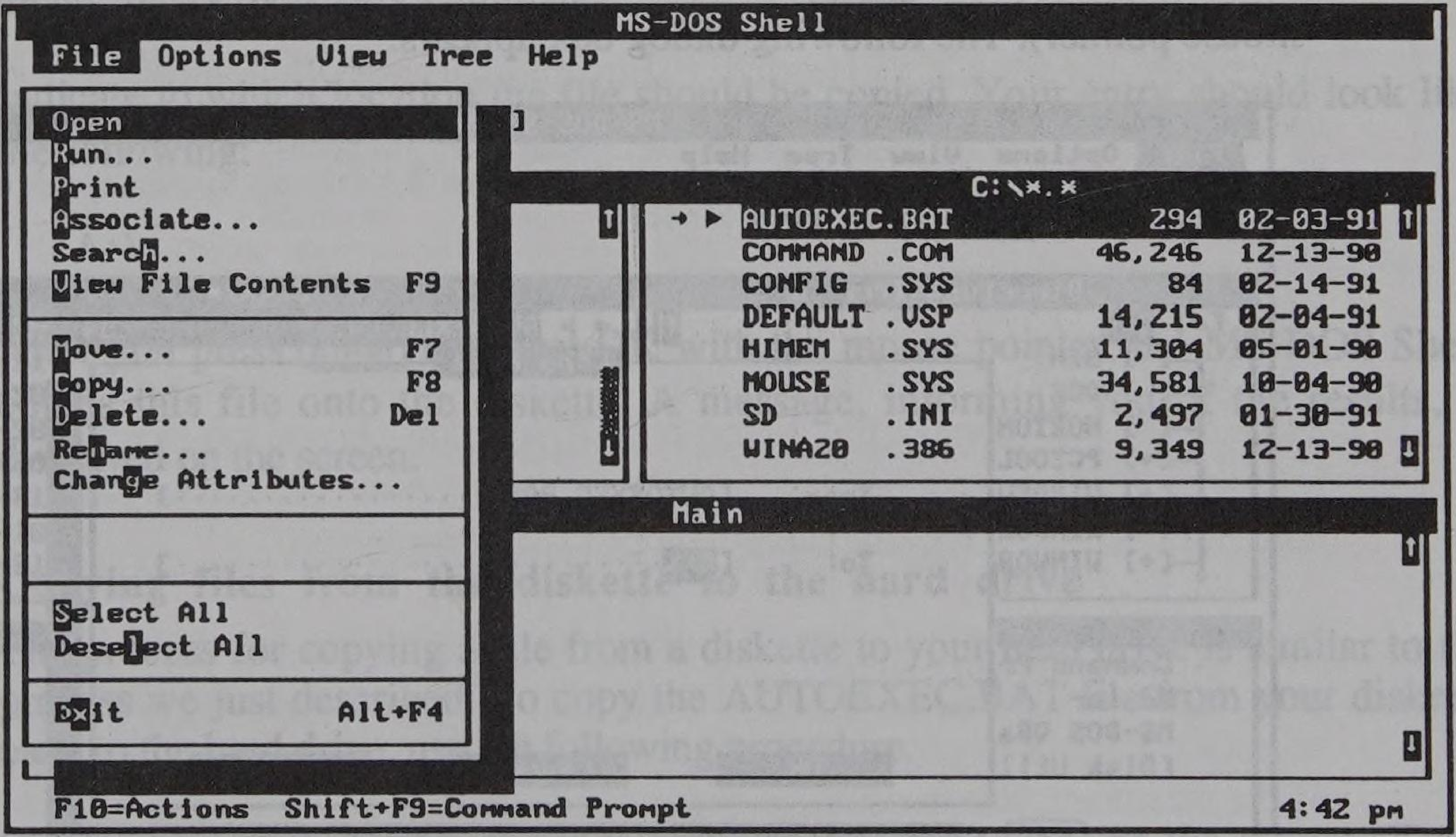
Suppose that you want to copy the AUTOEXEC.BAT file from your hard drive to a diskette. (This file was selected because we are certain that you have this file on your computer system.) To copy this file, first display the files in the main directory on the file list. Use the Tab key or the mouse to move the cursor to the Directory Tree and place the cursor on the following entry:

```
[ ] C:\
```

Before copying a file, the file must be marked. Use the Tab key or the mouse to select the file list area and then move the cursor to the file you want copied. In this example, this is AUTOEXEC.BAT file. This file will be marked with an triangular shaped selection cursor.

The file is copied by selecting a command from one of the menus of the MS-DOS Shell.

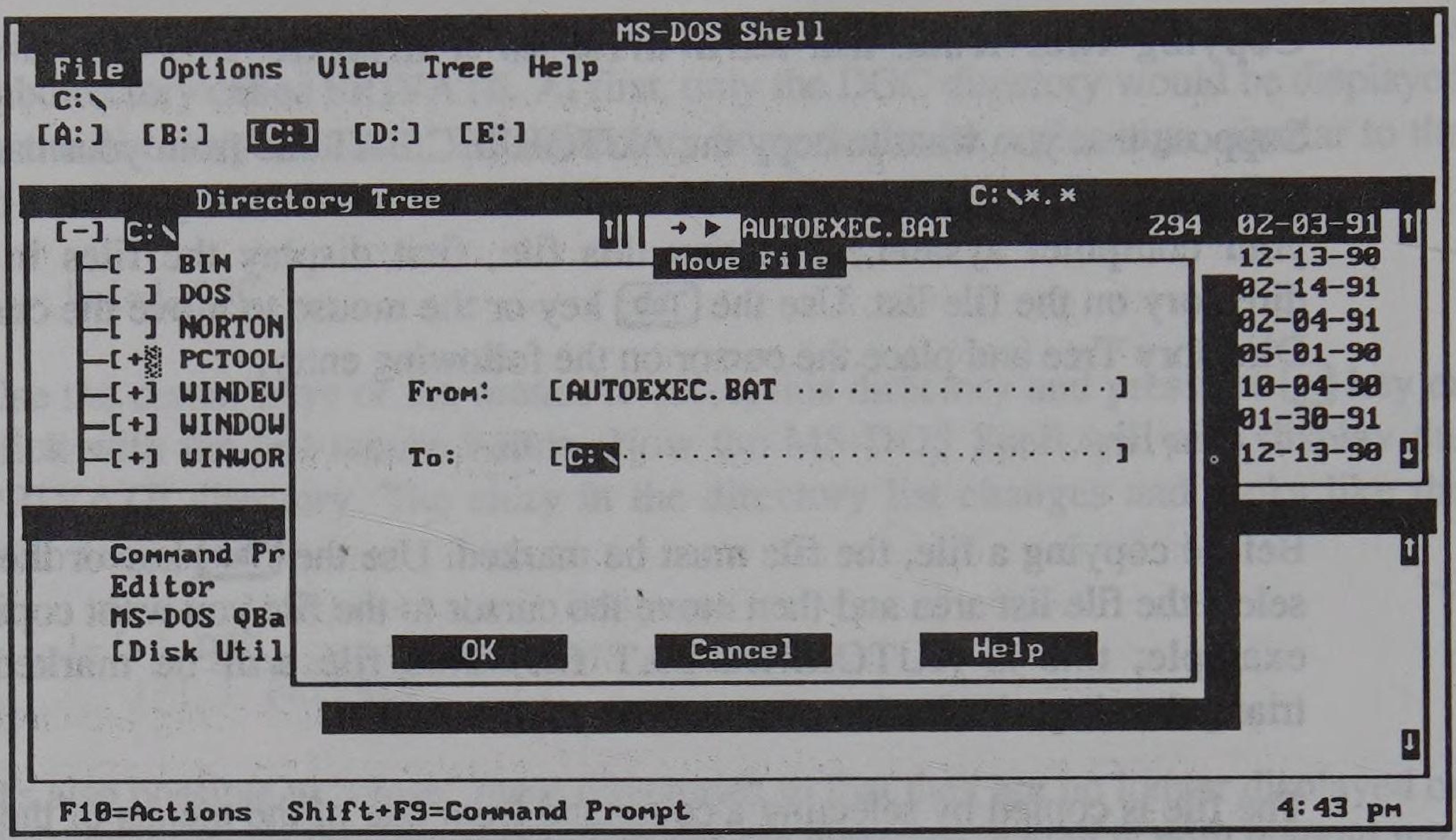
The MS-DOS Shell menus are selected by pressing Alt or by pointing and clicking with the mouse. Use the cursor keys or the mouse to select the desired menu and press Enter or the left mouse button to activate a menu.


MS-DOS Shell
File
Options
View
Tree
Help
Open
Run...
Print
Associate...
Search...
View File Contents F9
Move... F7
Copy... F8
Delete... Del
Rename...
Change Attributes...
Select All
Deselect All
Exit Alt+F4
C:\*.*
Main
F10=Actions
Shift+F9=Command Prompt
4:42 pm

```
→ ► AUTOEXEC.BAT        294  02-03-91
    COMMAND  .COM     46,246  12-13-90
    CONFIG   .SYS         84  02-14-91
    DEFAULT  .VSP     14,215  02-04-91
    HIMEM    .SYS     11,304  05-01-90
    MOUSE    .SYS     34,581  10-04-90
    SD       .INI      2,497  01-30-91
    WINA20   .386      9,349  12-13-90
```

Commands available in an activated menu

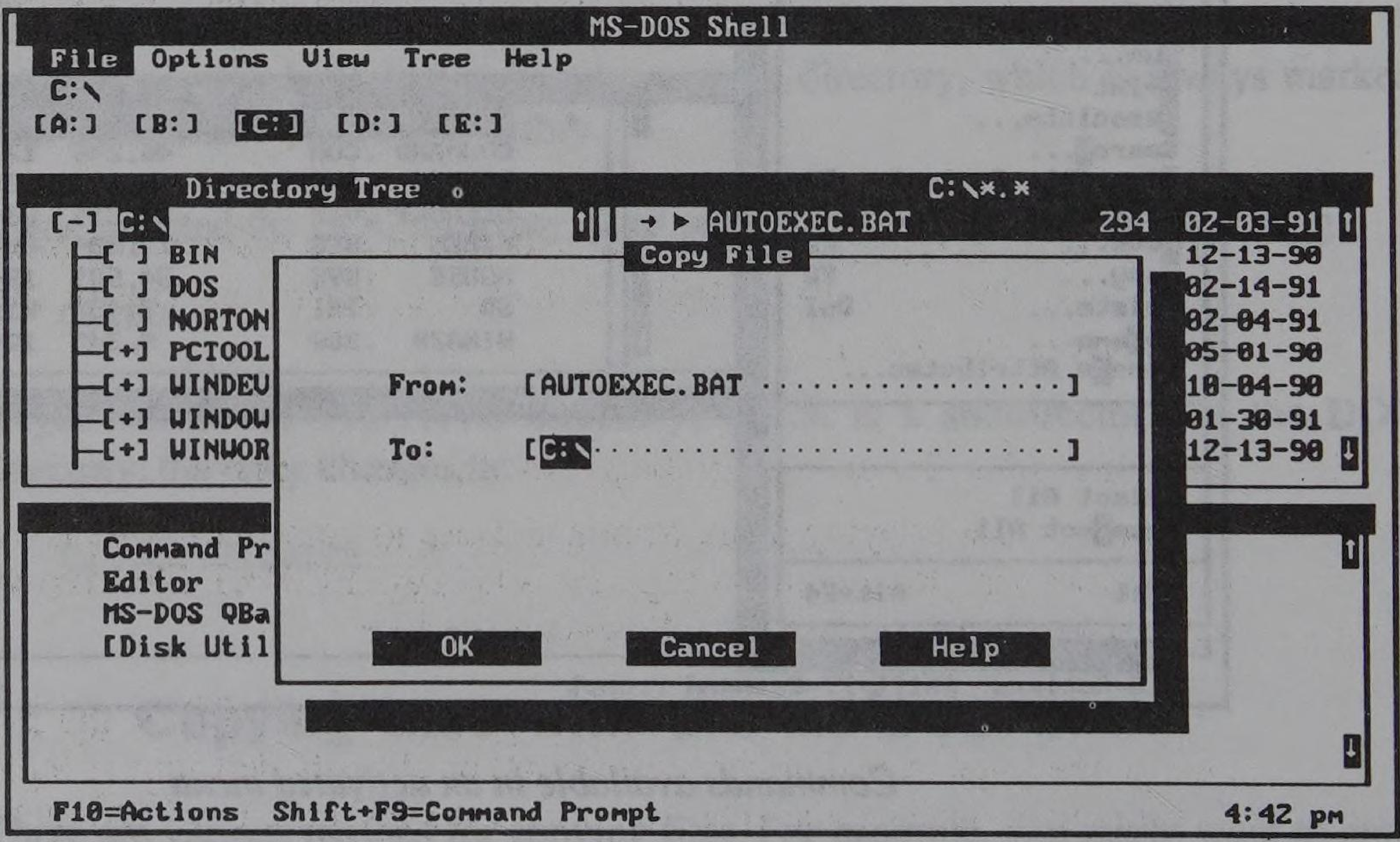
A pull-down menu containing various commands appears. To select these commands, use the cursor keys and to activate them press Enter. You can also use the mouse to point to the command and then click the left mouse button. Depending on the selected command, either execution begins immediately or a dialog box asking for more information appears.


MS-DOS Shell
File Options View Tree Help
C:\
[A:] [B:] [C:] [D:] [E:]
Directory Tree
C:\*.*
[-] C:\
[ ] BIN
[ ] DOS
[ ] NORTON
[+] PCTOOL
[+] WINDEV
[+] WINDOW
[+] WINWOR
AUTOEXEC.BAT 294 02-03-91
12-13-90
02-14-91
02-04-91
05-01-90
10-04-90
01-30-91
12-13-90
Command Pr
Editor
MS-DOS QBa
[Disk Util
Move File
OK
Cancel
Help
F10=Actions Shift+F9=Command Prompt
4:43 pm

```
From:    [AUTOEXEC.BAT ..................]
To:      [C:\............................]
```

A dialog box

To copy the file, use (Alt) to select the menu and press (Enter) (or select it with the mouse) to activate the **File** menu. Now use the cursor keys to select the **Copy** command and press (Enter) a second time (or select the **Copy** command with the mouse pointer). The following dialog box appears.


MS-DOS Shell
File Options View Tree Help
C:\
[A:] [B:] [C:] [D:] [E:]
Directory Tree
C:\*.*
[-] C:\
[ ] BIN
[ ] DOS
[ ] NORTON
[+] PCTOOL
[+] WINDEV
[+] WINDOW
[+] WINWOR
AUTOEXEC.BAT 294 02-03-91
12-13-90
02-14-91
02-04-91
05-01-90
10-04-90
01-30-91
12-13-90
Command Pr
Editor
MS-DOS QBa
[Disk Util
Copy File
OK
Cancel
Help
F10=Actions Shift+F9=Command Prompt
4:42 pm

```
From:    [AUTOEXEC.BAT ..................]
To:      [C:\............................]
```

The Copy dialog box

In this dialog box, enter the target drive (drive A:) and the directory. In our example we want to copy the file to the root directory. Type the following:

```
A:\
```

After entering the target drive, press Enter. The MS-DOS Shell copies the file and displays a message informing you of the copy operation. This message may appear only briefly, depending on the size of the file.

Change drives by using Tab or the mouse to select the letter A for the disk drive and press Enter. The file list area displays the contents of the diskette. The AUTOEXEC.BAT file is displayed on the diskette.

Now change back to the hard drive using Tab or the mouse.

You have just copied a file from the main directory of the hard drive onto the diskette. Now let's copy a file called FORMAT.COM, which is in the DOS directory. This is just as easy to do. Make the DOS directory the current directory by using Tab to move the cursor to the directory list and using the cursor keys (or the mouse pointer) to select the DOS directory. The path specification now displays the current directory.

Copy the file, as we described above, by using Tab or the mouse to go to the file list area, mark the file you want to copy and then select the command for copying. To do this, press Alt to activate the menu, use the cursor to choose the **File** menu and press Enter or use the mouse to make your selection. In the pull-down menu, select the **Copy** command.

Indicate to which location the file should be copied. Your entry should look like the following:

```
A:\
```

After you press Enter or select OK with the mouse pointer, the MS-DOS Shell copies this file onto the diskette. A message, informing you of the results, is displayed on the screen.

Copying files from the diskette to the hard drive

The process for copying a file from a diskette to your hard drive is similar to the process we just described. To copy the AUTOEXEC.BAT file from your diskette back to the hard drive, use the following procedure.

Choose drive A: by activating the drive icons with Tab or the mouse. Then choose the letter A and either press Enter or click the left mouse button. The file list area now displays the files that are on the diskette. Use Tab or the mouse to choose the file list and mark the AUTOEXEC.BAT file with the cursor keys or the mouse.

You can use Alt or the mouse to activate the menu bar, or the mouse to select the **File** menu. Next, select the **Copy** command. Enter the target in the dialog box. In this example we want to copy to the main directory of the hard drive. Type the following:

```
C:\
```

After pressing Enter or selecting OK with the mouse, a message will inform you that the file has been copied.

Next we'll copy the LETTER.DOC file, from the PRIVATE directory on the companion diskette, to drive C. Use Tab or the mouse to select the drive icons and choose the letter A for the drive. Then select the DOC directory in the Directory Tree. Press + to open the DOC directory, then select the PRIVATE directory. Now activate the file list area with Tab and use the cursor keys or use the mouse to mark the LETTER.DOC file.

Use Alt or the mouse to activate the menu bar and select the **File** menu. Choose the **Copy** command. In the dialog box enter the target drive for the copy operation.

Enter the entire path specification:

```
C:
```

When you press Enter, the MS-DOS Shell copies the file to the specified directory.

When copying files, it's important to make the drive and directory, which contains the file you want to copy, the current drive and directory. Then mark the file in the file list. Finally, in the dialog box, specify where the file should be copied.

Displaying pathnames

To copy a file to a specific directory of the hard drive, always specify the exact name of the directory in the dialog box. If you don't know the exact name, use the following method to find it. Remember, always enter the exact pathname of the current directory in the line directly above the drive icons.

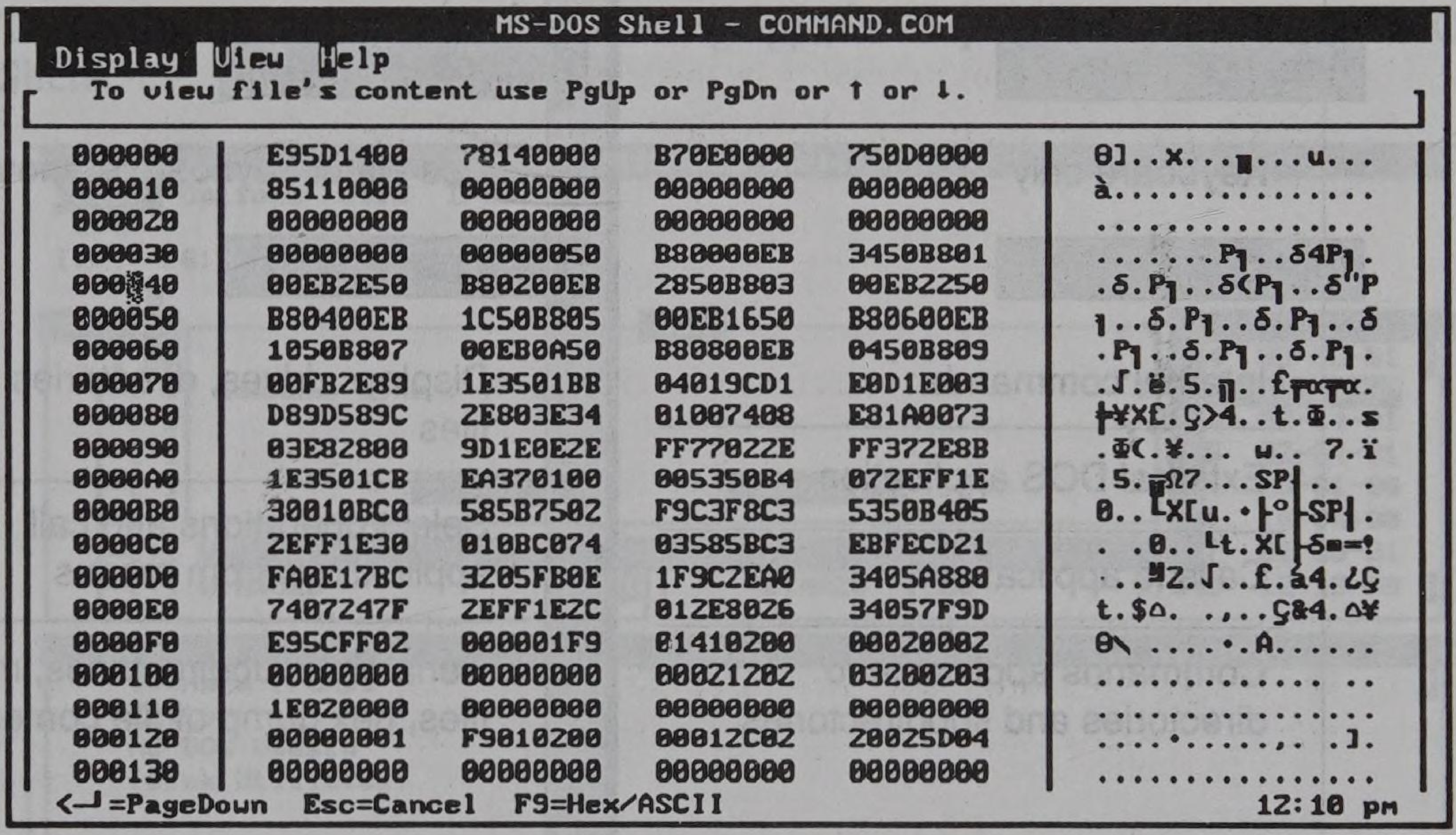
To determine the exact name of a directory, display the directory structure of the hard drive. Use Tab or the mouse to move the cursor to the directory list and select the target directory. In the line above the drive icons you'll see the pathname of the directory. When copying, enter this pathname in the dialog box.

1.3.3 Using the MS-DOS Shell to view files

Up to this point we've discussed the contents of directories and how to copy files. Now we'll discuss files and their contents in detail.

Displaying the contents of most of the files won't provide much information about them. Although the MS-DOS Shell provides the **View File Contents** command in the **File** menu, most files cannot be displayed with this command. First mark the file to be displayed. Suppose that you want to view the contents of the COMMAND.COM file located in the root directory of drive C:. As we previously discussed, make the root directory the current directory, then use Tab or the mouse to select the file list area and choose COMMAND.COM. Now activate the menu bar and choose the **File** menu. Use either the cursor keys or the mouse to select the **View File Contents** command.

The screen's appearance will change significantly. It will look similar to the following:



```
  To view file's content use PgUp or PgDn or ↑ or ↓.

  000000   E95D1400   78140000   B70E0000   750D0000   θ]..x...▮...u...
  000010   85110000   00000000   00000000   00000000   à...............
  000020   00000000   00000000   00000000   00000000   ................
  000030   00000000   00000050   B80000EB   3450B801   ........P╕..δ4P╕.
  000040   00EB2E50   B80200EB   2850B803   00EB2250   .δ.P╕..δ(P╕..δ"P
  000050   B80400EB   1C50B805   00EB1650   B80600EB   ╕..δ.P╕..δ.P╕..δ
  000060   1050B807   00EB0A50   B80800EB   0450B809   .P╕..δ.P╕..δ.P╕.
  000070   00FB2E89   1E3501BB   04019CD1   E0D1E003   .√.ë.5.╗..£╤α╤α.
  000080   D89D589C   2E803E34   01007408   E81A0073   ╪¥X£.Ç>4..t.Φ..s
  000090   03E82800   9D1E0E2E   FF77022E   FF372E8B   .Φ(.¥... w.. 7.ï
  0000A0   1E3501CB   EA370100   005350B4   072EFF1E   .5.╦Ω7...SP┤.. .
  0000B0   30010BC0   585B7502   F9C3F8C3   5350B405   0..└X[u.·├°├SP┤.
  0000C0   2EFF1E30   010BC074   03585BC3   EBFECD21   . .0..└t.X[├δ■═!
  0000D0   FA0E17BC   3205FB0E   1F9C2EA0   3405A880   ...╝2.√..£.á4.¿Ç
  0000E0   7407247F   2EFF1E2C   012E8026   34057F9D   t.$⌂. .,..Ç&4.⌂¥
  0000F0   E95CFF02   000001F9   01410200   00020002   θ\ ....·.A......
  000100   00000000   00000000   00021202   03200203   ............. ..
  000110   1E020000   00000000   00000000   00000000   ................
  000120   00000001   F9010200   00012C02   20025D04   ....·.......,. .].
  000130   00000000   00000000   00000000   00000000   ................
<─┘=PageDown  Esc=Cancel  F9=Hex/ASCII                          12:10 pm
```

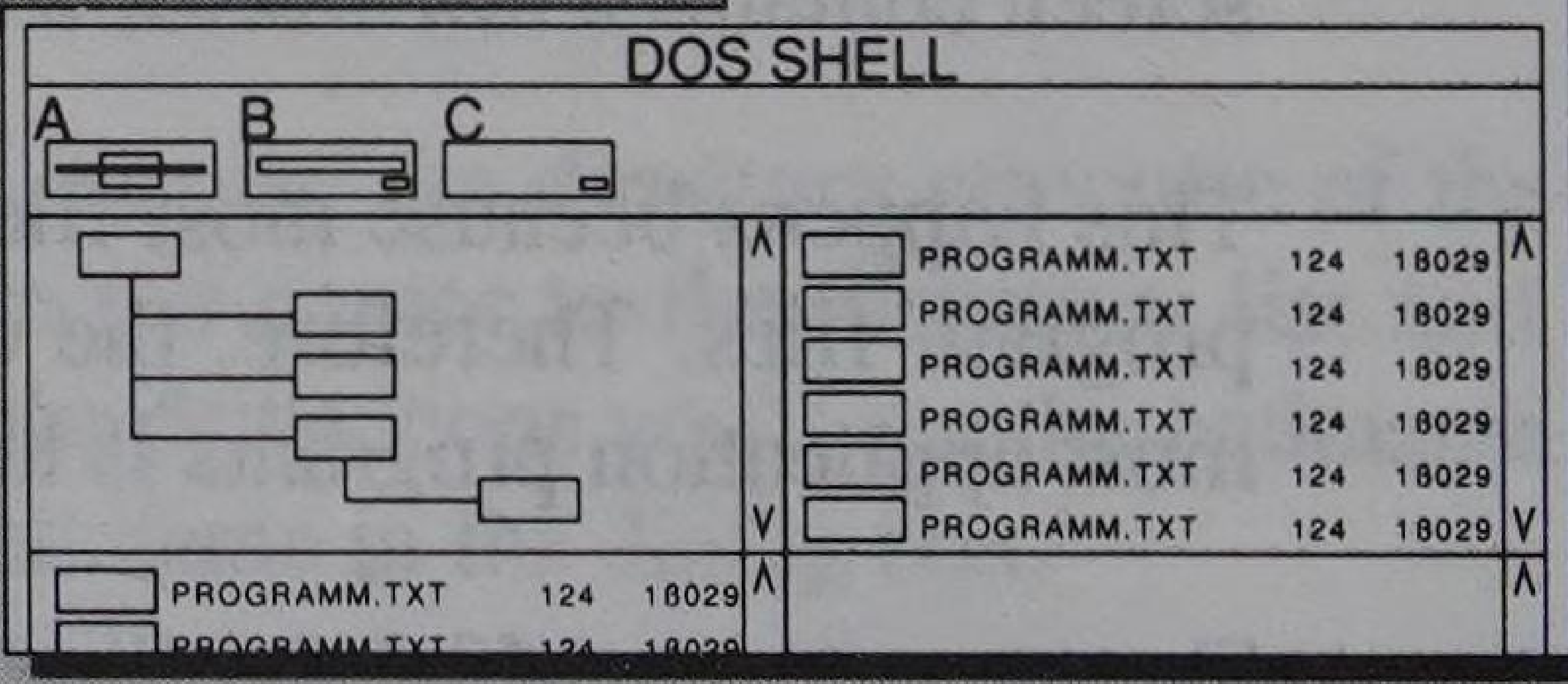
The contents of the COMMAND.COM file

As you can see, this screen doesn't provide much useful information about the file. The left half of the screen contains numbers and letters and the right half of the screen contains a few words, which you may recognize.

This happens because most files are linked to certain application programs or are program files. Therefore, the only way to understand files that are created with most application programs is to use them with the appropriate programs.

To return to the MS-DOS Shell, press Esc.

However, some files can be very useful when they're displayed. These files are usually ASCII text files, such as the AUTOEXEC.BAT file. To display this file, which is usually located in the root directory of drive C, select it. Activate the menu and select the **File** menu to display the menu commands. Next, select the **View File Contents** command.

	Command line interface C:>	DOS–Shell
Application	This is of limited help to beginners: better suited for more advanced user	User-friendly interface best for beginners
Access	Keyboard only	Use either keyboard or mouse
Options	Internal commands	Displays drives, directories and files
	External DOS application	Select operations and call applications from menus
	Calls to application	Renaming subdirectories, moving files, hex dump of file contents
	Commands applicable to directories and subdirectories	
Sample screen	C:\>	

DOS SHELL
A
B
C
PROGRAMM.TXT 124 18029
PROGRAMM.TXT 124 18029
PROGRAMM.TXT 124 18029
PROGRAMM.TXT 124 18029
PROGRAMM.TXT 124 18029
PROGRAMM.TXT 124 18029
PROGRAMM.TXT 124 18029

MS–DOS 5.0 is presented in two forms:

- Standard installation
- Include reciprocal calls

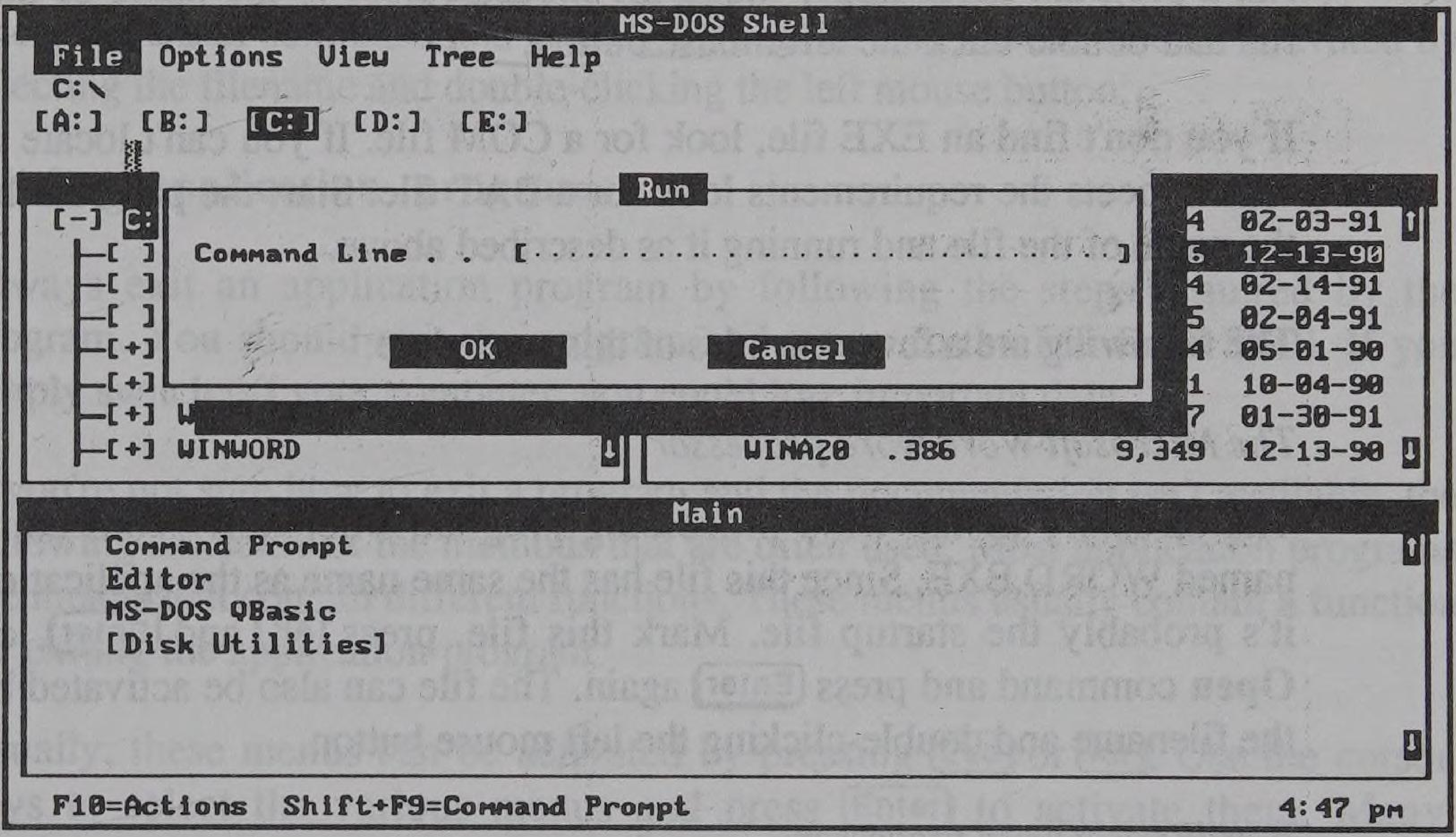
A comparison between the command line interface and MS-DOS Shell

1.3.4 Running programs from the MS-DOS Shell

Usually it's necessary to use the application program that created the files in order to read the files' contents. So, to display a word processor file, start the word processor used to create the file and to display a file containing data records, start the database which was used to create the data records.

The ability to use various applications is the main reason for purchasing a computer. So, besides being used to display existing files, these application programs are also used to create new files.

The following are a few tips on running application programs from the MS-DOS Shell.


MS-DOS Shell
File Options View Tree Help
C:\
[A:] [B:] [C:] [D:] [E:]
Run
Command Line . . [......................................]
OK
Cancel
[-] C:
[ ]
[ ]
[ ]
[+]
[+]
[+] W
[+] WINWORD
4 02-03-91
6 12-13-90
4 02-14-91
5 02-04-91
4 05-01-90
1 10-04-90
7 01-30-91
WINA20 .386 9,349 12-13-90
Main
Command Prompt
Editor
MS-DOS QBasic
[Disk Utilities]
F10=Actions Shift+F9=Command Prompt
4:47 pm

Running applications from the MS-DOS Shell

Starting programs

The files created with an application program are stored in the program's own directory. So, to start an application program, first select the directory in which these files are located.

Then run a specific file, called the startup file. The name of this file is often identical to the name of the program. If the name of the program is longer than eight letters, the startup filename will consist of either a part of the name or an abbreviation of the program name. (MS-DOS limits filenames to eight characters, plus a three letter filename extension.)

In order to start an application program, find the startup file for the program. This file can be identified by its filename extension, which could be one of the following extensions:

```
.EXE
.COM
.BAT
```

To find this file, display the contents of the directory in which the program is located. Look for files that have the .EXE extension and that also meet the requirements for a program name. If you find such a file, you can try to start it by marking it and using Alt and Enter to select the **File** menu. Then choose the **Open** command and press Enter again. You can also use the mouse to select and run a program file. Simply move the mouse cursor to the name of the file to be run and double-click the left mouse button.

If you don't find an EXE file, look for a COM file. If you can't locate a COM file which meets the requirements look for a BAT file. Start the program by selecting the name of the file and running it as described above.

The following are a few examples of this procedure:

The Microsoft Word word processor

The directory of the word processing program Microsoft Word contains a file named WORD.EXE. Since this file has the same name as the application program, it's probably the startup file. Mark this file, press Alt and Enter, choose the **Open** command and press Enter again. The file can also be activated by selecting the filename and double-clicking the left mouse button.

WordPerfect word processor

The directory of WordPerfect contains a file named WP.EXE. Since this file has the same name as the application program, it's probably the startup file. Mark this file, press Alt and Enter, choose the **Open** command and press Enter again. The file can also be activated by selecting the filename and double-clicking the left mouse button.

Lotus 1-2-3 spreadsheet

The directory of Lotus 1-2-3 contains a file named LOTUS.EXE. Since this file has the same name as the application program, it's probably the startup file. Mark this file, press Alt and Enter, choose the **Open** command and press Enter again. The file can also be activated by selecting the filename and double-clicking the left mouse button.

dBASE IV data management system

The directory of dBASE IV contains a file named DBASE.EXE. Since this file has the same name as the application program, it's probably the startup file. Mark this file, press Alt and Enter, choose the **Open** command and press Enter again. The file can also be activated by selecting the filename and double-clicking the left mouse button.

WINDOWS

There are many files with the .EXE extension in the WINDOWS directory. None of these files meet the requirement for the name of a startup file. However, there is a WIN.COM file which meets both requirements for the name of the startup file. Mark this file and start it by pressing Alt and Enter. In the menu that appears, select the **Open** command and press Enter again. The file can also be activated by selecting the filename and double-clicking the left mouse button.

Exiting application programs

Always exit an application program by following the steps required by the program. You should quit the program and return to the MS-DOS Shell. If you simply switch off your computer, you could lose important data.

If you're not sure how to exit a program and the documentation isn't available, the following are some of the methods that are often used. Most application programs use menus that contain different functions. These menus usually contain a function for exiting the application program.

Usually, these menus can be activated by pressing Esc or Alt. Use the cursor keys to select the various menus and press Enter to activate them. Many programs also support the use of a mouse.

The following commands are used for exiting most application programs:

QUIT, END, LEAVE, or something similar.

Choose the appropriate command and press Enter. Many programs display a security prompt, which asks you to confirm whether or not you really want to leave the program.

The MS-DOS Shell

To exit the MS-DOS Shell, press Alt F, then select the **Exit** command and press Enter. You'll be returned to the MS-DOS command line.

WINDOWS

Press Alt and then F to select the command, **Exit Windows** and press Enter. A security prompt, asking whether or not you really want to end the program, will appear. Press Enter to return to the MS-DOS command line.

☞ For complete instructions on exiting a program, you should refer to the application program's users manual. If you exit a program incorrectly, important data could be lost.

1.3.5 Deleting files using the MS-DOS Shell

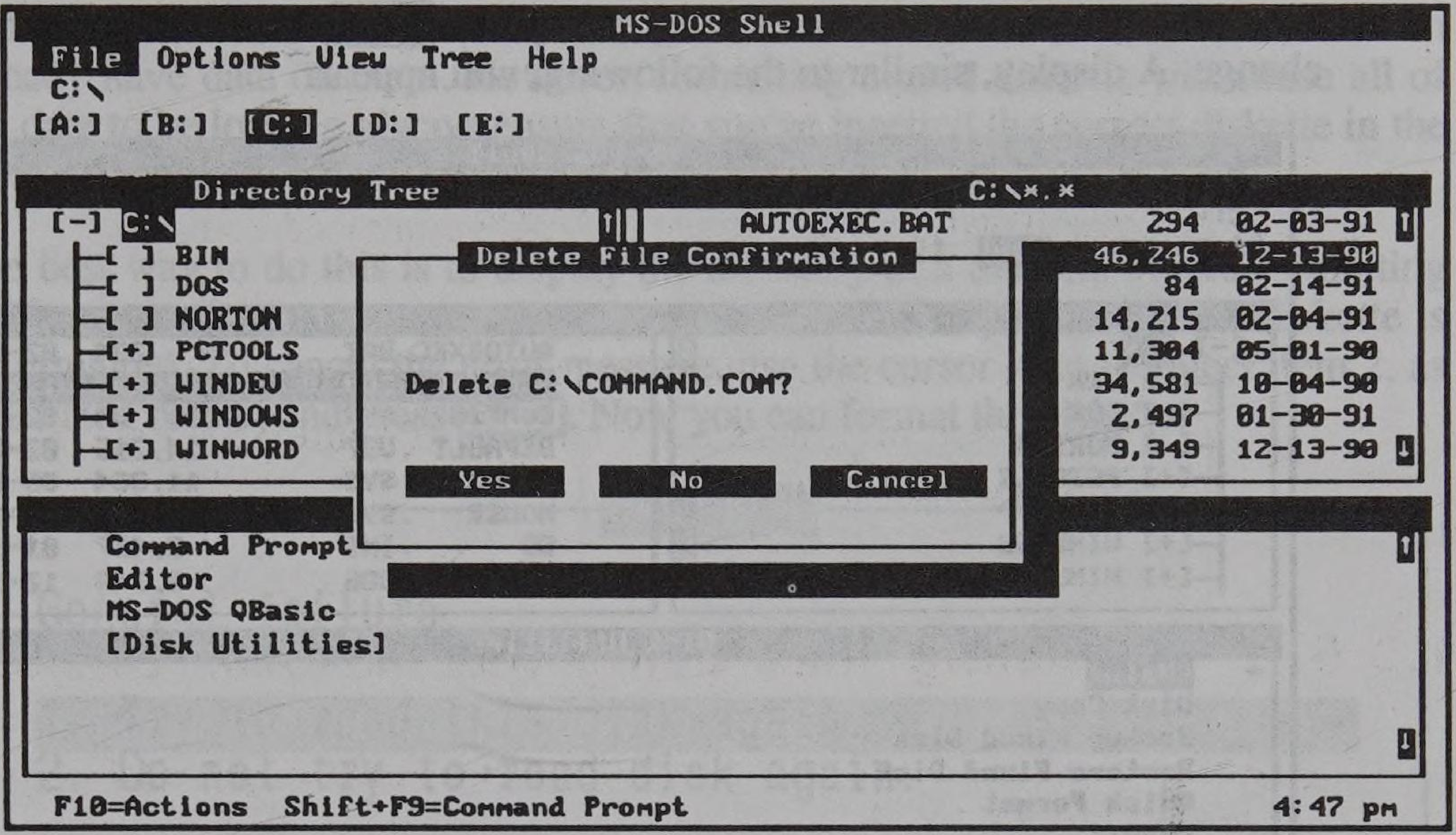
To remove an old file from the hard drive or a diskette, use the **Delete** command from the **File** menu.

Be very careful when deleting files. Even though MS-DOS 5.0 includes an undelete command, generally once a file is deleted, it's permanently gone. So before deleting a file, make sure that you no longer need it.

Before deleting a file, you must mark it. Make the drive, which contains the file, the current drive and select the directory containing the file. Then use Tab to move to the file list area and move the cursor to the file you want to delete.

Select the menu bar by pressing Alt and press Enter to activate the **File** menu. Use the cursor keys to select the **Delete** command and press Enter again. DOS will display a security prompt, which must be confirmed before the file is deleted. Confirm the deletion by pressing Enter.

The selected file will be deleted and a message informing you of this, will appear. If you discover that the wrong file has been selected or that you really don't want to delete the selected file, it's possible to cancel the deletion during the security prompt by pressing Esc.


MS-DOS Shell
File Options View Tree Help
C:\
[A:] [B:] [C:] [D:] [E:]
Directory Tree
C:\*.*
[-] C:\
[ ] BIN
[ ] DOS
[ ] NORTON
[+] PCTOOLS
[+] WINDEV
[+] WINDOWS
[+] WINWORD
Delete File Confirmation
Yes
No
Cancel
Command Prompt
Editor
MS-DOS QBasic
[Disk Utilities]
F10=Actions Shift+F9=Command Prompt
4:47 pm

```
AUTOEXEC.BAT      294  02-03-91
               46,246  12-13-90
                   84  02-14-91
               14,215  02-04-91
               11,304  05-01-90
               34,581  10-04-90
                2,497  01-30-91
                9,349  12-13-90
```

Delete C:\COMMAND.COM?

A security prompt appears before MS-DOS Shell deletes a file

Let's delete the LETTER.DOC file from drive A:. First make drive A: the current drive. Move the cursor to the drive icons, choose drive letter A: and press Enter. The file list then displays the contents of the diskette. Use Tab to move the cursor to the file list area. Mark the LETTER.DOC file with the cursor keys.

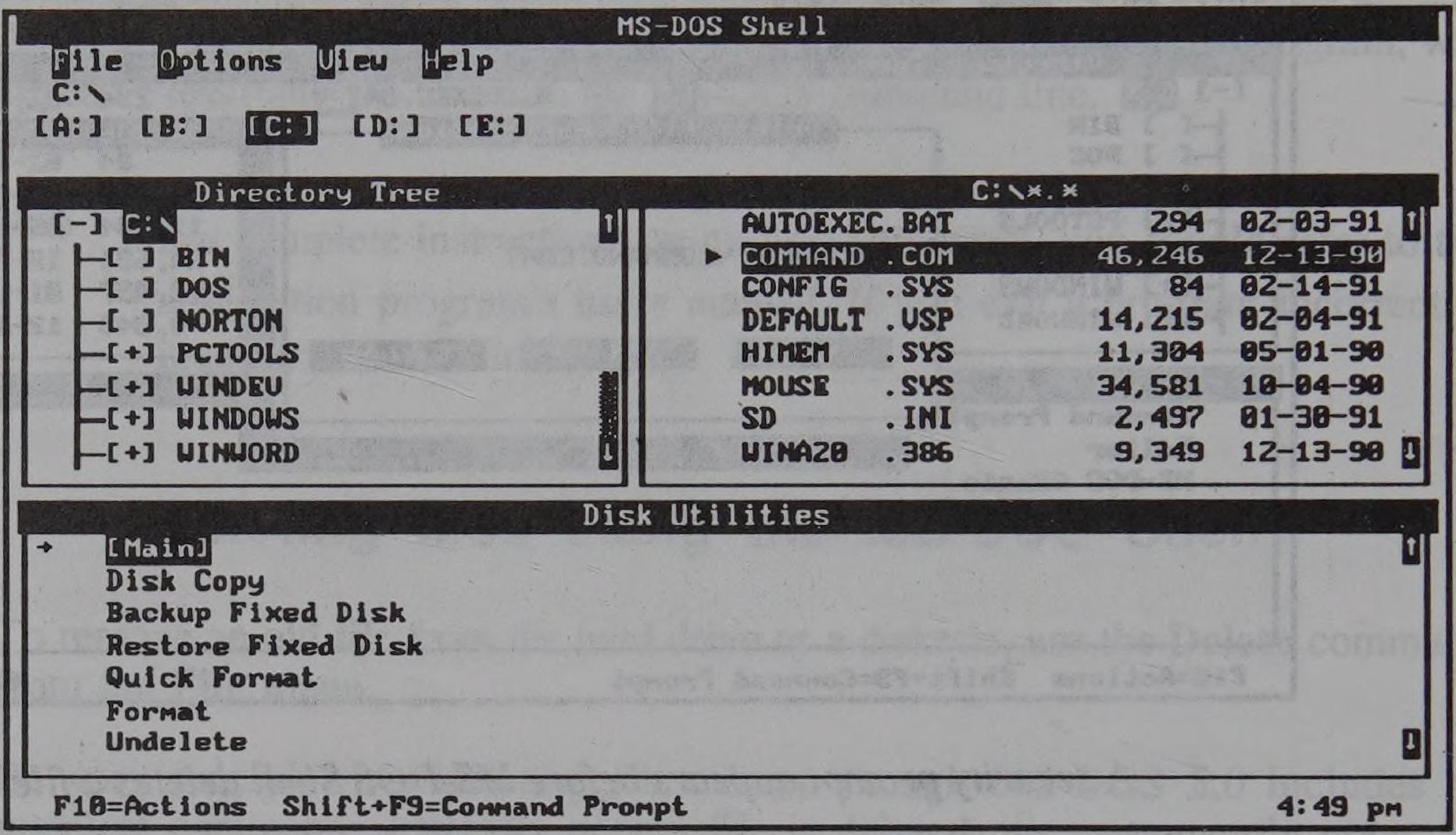
Next, use Alt to select the menu bar, choose the **File** menu and press Enter. In the pull-down menu, use the cursor keys to move to **Delete** and press Enter. Confirm the security prompt with Enter. The screen will display a message informing you that the file has been deleted. These selections can also be made by using a mouse.

When you delete files, it's important that the drive and directory which contain the file to be deleted is the current path. This will help prevent errors. Mark the file in the file list and select the command to delete. Before the file is deleted, a security prompt will always appear. Either confirm the prompt by pressing Enter or cancel the deletion process by pressing Esc.

1.3.6 Formatting diskettes with the MS-DOS Shell

Before new diskettes can be used with your computer, they must be formatted. MS-DOS will return an error message if you try to use an unformatted diskette. The process of preparing a disk for use is called "formatting the disk". To do this, use the **Format** command. In the MS-DOS Shell, this command is located in the Disk Utilities manager. Some useful disk utilities are built into the MS-DOS Shell and provide various tools for managing disks.

To format a diskette, press the Tab key until you're in the area that contains the Main heading. This area displays various programs supplied by MS-DOS. Move the cursor to the Disk Utilities entry and press Enter. The contents of this area change. A display, similar to the following, will appear:



```
                              MS-DOS Shell
 File  Options  View  Help
 C:\
 [A:]  [B:]  [C:]  [D:]  [E:]
          Directory Tree                            C:\*.*
 [-] C:\                            AUTOEXEC.BAT        294  02-03-91
     ├─[ ] BIN                    ▶ COMMAND  .COM    46,246  12-13-90
     ├─[ ] DOS                      CONFIG   .SYS        84  02-14-91
     ├─[ ] NORTON                   DEFAULT  .VSP    14,215  02-04-91
     ├─[+] PCTOOLS                  HIMEM    .SYS    11,304  05-01-90
     ├─[+] WINDEV                   MOUSE    .SYS    34,581  10-04-90
     ├─[+] WINDOWS                  SD       .INI     2,497  01-30-91
     └─[+] WINWORD                  WINA20   .386     9,349  12-13-90
                              Disk Utilities
 →   [Main]
     Disk Copy
     Backup Fixed Disk
     Restore Fixed Disk
     Quick Format
     Format
     Undelete
 F10=Actions   Shift+F9=Command Prompt                              4:49 pm
```

Disk Utilities

Among the commands displayed is the **Format** command. Use the cursor keys to select this command and press Enter. A dialog box then prompts you for the location of the drive that contains the diskette to be formatted. The default entry is A:. Press Enter to format a diskette in this drive.

☞ A mouse can also be used to select the commands for formatting a disk.

The MS-DOS Shell screen disappears and a message prompting you to place a diskette in drive A: appears on the screen. Once you press Enter the computer starts to format the disk. A message on the screen keeps you informed of the formatting process. The amount of time this process takes depends on your computer.

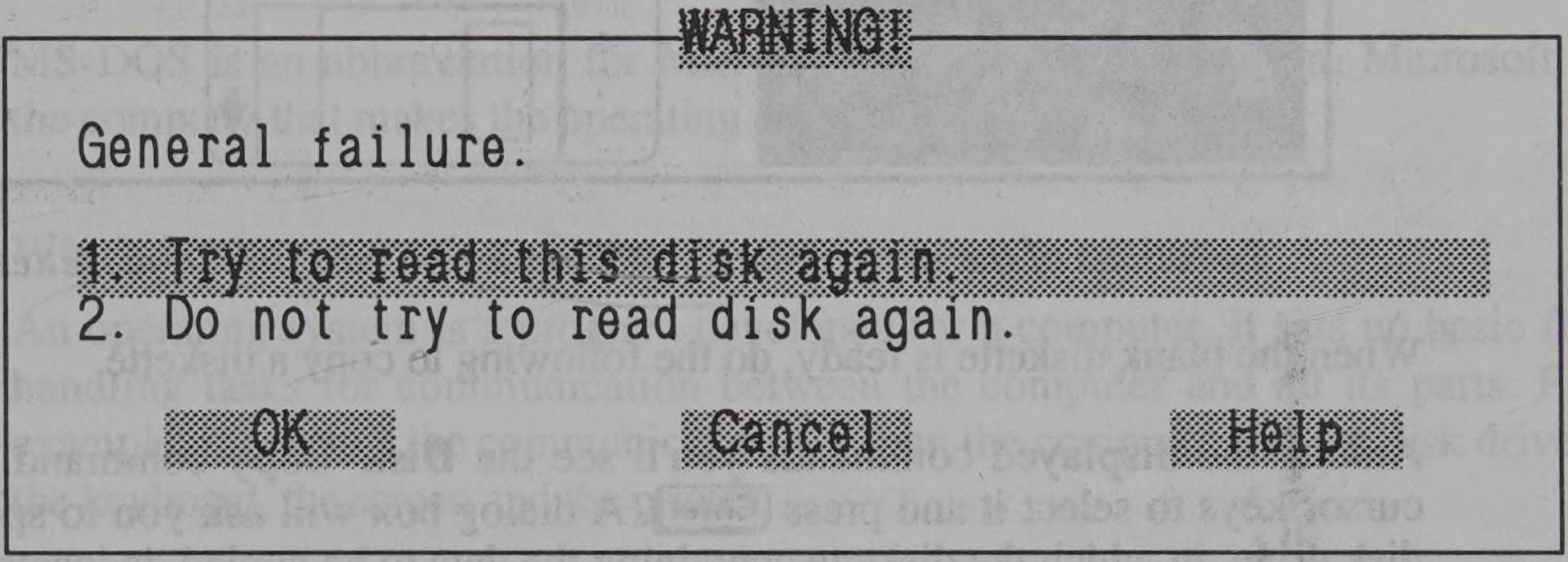
After the computer finishes formatting, you're prompted to give the diskette an optional volume name. A name of up to eleven characters may be entered. If you don't want to name the diskette, simply press Enter.

Some information about the disk, such as the amount of available storage space, is displayed on the screen.

The next message asks whether you want to format another diskette. If you do, press Y. You'll be prompted to insert another diskette and the entire process will be repeated. If you don't want to format another diskette, press N. This ends the **Format** command. You'll be prompted to press any key to return to the MS-DOS Shell. After doing this, the MS-DOS Shell screen appears.

Be careful when formatting diskettes. You can reformat used diskettes which already have data on them. However, formatting a used diskette will cause all of the data to be lost. So always ensure that you've inserted the correct diskette in the drive before starting the formatting procedure.

The best way to do this is to display the directory of a diskette before formatting it. If you receive an error message, you can be relatively sure that the diskette is unformatted. To cancel the error message, use the cursor keys to select item 2, as illustrated below, and press Enter. Now you can format the diskette.



```
WARNING!

General failure.

1. Try to read this disk again.
2. Do not try to read disk again.
```

Non-formatted disk error message

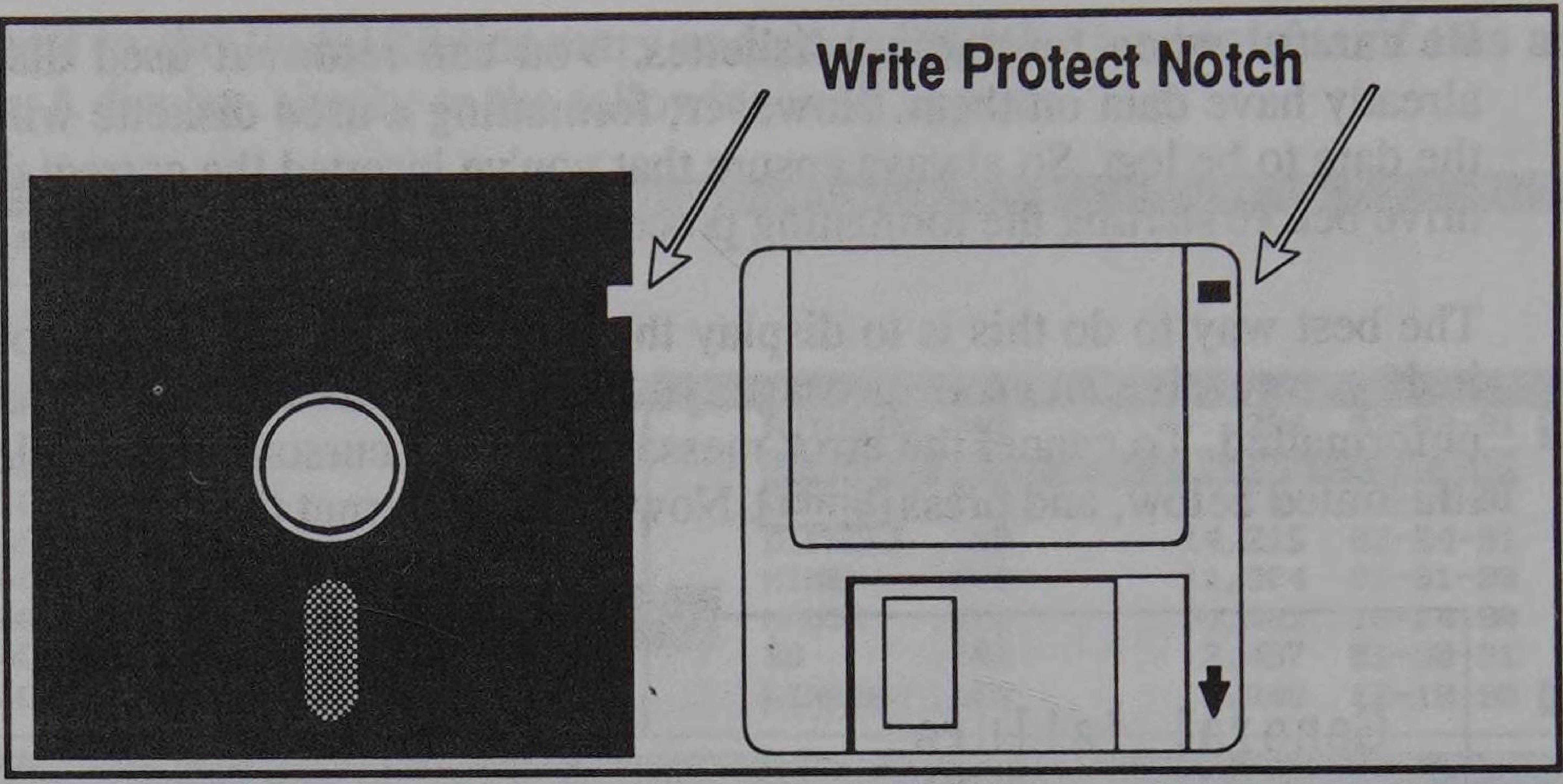
1.3.7 Copying diskettes with the MS-DOS Shell

Occasionally you'll have to copy entire diskettes. For example, you might want to create backup copies of your program diskettes or you might want to send someone a disk containing all or part of the files on a particular diskette.

To copy a diskette, you'll need a blank diskette. Like all functions from the MS-DOS Shell, a mouse with an installed mouse driver can be used to make the selections. Although this is the easiest method to use, we'll describe the alternative method, which involves using the keyboard to make selections.

Before copying diskettes, you should write protect the diskettes in order to protect the original diskette. For a 5 1/4" diskette, cover the write protect notch with an adhesive tab. For a 3 1/2" diskette, slide the small tab so that the hole is exposed.

Use the **Disk Copy** command, which is part of the Disk Utilities program under the Main section of the MS-DOS Shell screen. To select this command, press Tab until you are in the area with the Main heading. In this area you'll see various programs that are supplied by MS-DOS. Move the cursor to the Disk Utilities entry and press Enter.



Locations of the write protect notch on 5 1/4" and 3 1/2" diskettes

When the blank diskette is ready, do the following to copy a diskette.

Among the displayed commands you'll see the **Disk Copy** command. Use the cursor keys to select it and press Enter. A dialog box will ask you to specify the disk drive, in which the diskette containing the data to be copied, is located. You'll also be asked to indicate the drive in which the target diskette (the diskette to which the data will be copied) is located. The default entry is A: B:. If you only have one disk drive, you must change this line as follows:

```
A: A:
```

Even if you have two disk drives, you should change the copying commands to the same disk drive. Various conditions, which we won't discuss here, must be met before the copying process is successful. So, to ensure that the diskette is copied properly, follow the procedure as described even if you have two disk drives.

The MS-DOS Shell screen disappears and a message asks you to insert the diskette, to be copied (the source diskette), in drive A: and press any key. Once you do this, a message indicates that the diskette is being copied and provides some specific information about the copy.

After a short time, you'll be prompted to insert the target diskette into drive A:. Insert the blank diskette in the disk drive and press any key. If you're copying diskettes with a memory capacity of 1.2 Meg, the diskettes must be swapped several times. A message will indicate when to swap the diskettes.

After the entire diskette has been copied, a dialog box will ask whether you want to copy another diskette. If you do, press Y and follow the prompts to repeat the copy process. If you don't want to copy another diskette, simply press N.

This terminates the **Disk Copy** command and returns you to the MS-DOS Shell.

1.4 What You Should Know About MS-DOS

This section is a review of the material we've discussed up to this point. We've tried to include many of the terms that you'll encounter as you work with computers.

1.4.1 What is MS-DOS?

MS-DOS is an abbreviation for Microsoft Disk Operating System. Microsoft is the company that makes the operating system for your computer.

What is an operating system?

An operating system is a program developed for a computer. It sets up basic file handling tasks for communication between the computer and all its parts. For example, it controls the communication between the computer and the disk drives, the keyboard, the screen and the printer.

One connection is between the keyboard and the screen. The operating system ensures that the characters you type on the keyboard actually appear on the screen.

So, the operating system provides a means of communicating with all of the devices to which your computer is connected.

The operating system also provides the user with the basic commands that are needed in order to work with these devices. Some of these commands, such as the **Copy**, **Delete** and **Format** commands, were discussed in the previous sections.

Although your computer needs an operating system in order to work, an operating system by itself can't create letters, manage databases, calculate numerical tables, etc. Application programs are used for these tasks. Word processors, databases and spreadsheets are examples of such application programs. These programs make your daily work on the computer much easier. You can use them to write letters or books, manage large databases or to do calculations for an entire company.

All of these application programs use the operating system's functions or services to help them perform their work. Obviously you can work with a computer that doesn't contain any application programs. You can copy, delete, format, etc. However, if you try to work on a computer that doesn't have an operating system or you try to use only an application program, you'll quickly discover that it won't work.

Although MS-DOS is only one of several different operating systems, it has made the personal computer a success. It is the most widely used operating system in the world.

Operating System Basic Tasks

System configuration	FILES; BUFFERS; COUNTRY;PROMPT; COMSPEC; DRIVERS ...
Storage media	Prepare/format Name/rename Copy/compare Check Display contents ...
Directories	Make/remove Change/rename Display contents ...
Files	Create/delete Copy/rename Print Display contents
Program	Find/start End (return to DOS) Memory management Storage media management

The operating system is the heart of every computer

1.4.2 The command line and the MS-DOS Shell

MS-DOS 5.0 provides two ways to work with your computer. Both of these options, the command line and the MS-DOS Shell, were introduced earlier in this book. Now we'll explain the differences between these methods in more detail.

The differences in operation

The previous sections highlighted the differences between using the command line and the MS-DOS Shell. The command line requires that you enter commands directly. This means that if you use only the command line, all of the DOS commands must be memorized or a reference book must be kept next to the computer so that you can look up the various commands and functions.

The MS-DOS Shell is unique. The screen display contains a lot of information. The commands and functions can be selected from menus displayed on the screen. So, instead of memorizing commands and functions, you can select and start operations from menus. In these menus, commands and functions, which are organized according to their functions, can be easily identified. The MS-DOS Shell is referred to as a user interface.

The difference between the command line and the MS-DOS Shell

There are a few other differences between the command line and the MS-DOS Shell. The MS-DOS Shell makes several options of the operating system easily accessible. However, you cannot perform all available functions of the operating system from the MS-DOS Shell. Many of these functions can only be used at the command line.

For example, the MS-DOS command CHKDSK can be used to check the integrity of your hard drive or diskette. However, this command can only be used from the command line. The MS-DOS Shell doesn't provide direct access to this command.

The command line allows you to use the operating system's functions. The MS-DOS Shell, however, allows you to use only the most basic and necessary functions. Although the MS-DOS Shell can be used for your daily work, the command line is necessary for special operations.

1.4.3 What are files, directories etc.

In this section we'll explain more terms that you'll encounter as you work with the computer.

What is a file?

In the previous sections we used the term *file* without explaining its meaning. Let's examine, in more detail, what files actually are.

How do you recognize a file

Every file has a filename, up to eight characters long, and possibly a file extension, up to three characters long. The file extension is separated from the filename by a period.

The files are always a specific size in bytes. It's possible to display this size and information about the file creation date.

Files are data sets that belong together. Files always fulfill a certain purpose. There are different kinds of files, divided into program files and data files.

Program files are needed in order to use the program. These files contain information, about the program's functions, that belong with the file. Without these program files, application programs wouldn't function. That's why every application program contains at least one of these files. You can always recognize a program file by its extension, either .EXE or .COM.

All other files are data files. These files are created, by users, with programs. Generally, these files can't be used without the program used to create them. For example, a word processor can be used to create files containing text. A database can create files containing associated groups of information, such as addresses. These files are always closely linked to the program that was used to create them. Ordinarily, you can only use data files effectively with the program that created them.

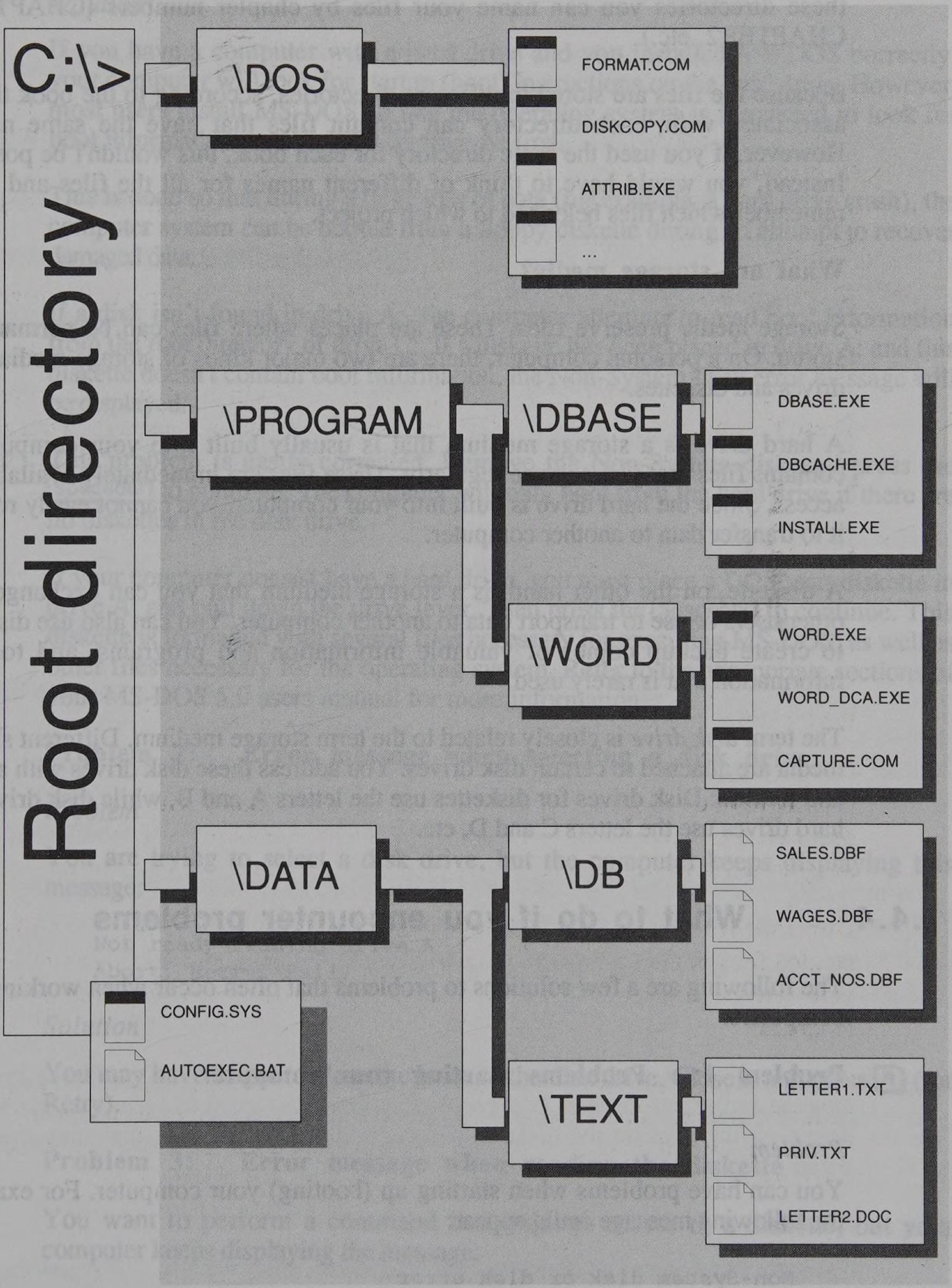
What are directories?

Directories create order on a hard drive or diskette. Every hard drive or diskette has a root directory, which is identified by a drive letter, a colon and a backslash. The root directory of the hard drive is C:\. From the root directory you can branch off to other directories, which contain other files.

A directory contains files that belong together. For example, different programs usually have directories containing files that are associated with the programs. All of these directories have a name. WORD and DBASE are examples of such programs. They always place files in their own directories when installed.

Directories are also used for storing created files. However, if you place all of the created files in the root directory, after a while it would contain so many files that you would lose track of them. So, remember that you should keep your hard drive organized.

Another reason for using directories is that the files in a directory must have different names. You cannot have two files with the same name in the same directory. After a while you will run out of names for files. However, directories can contain files with the same name as files stored in another directory.


Root directory
C:\>
\DOS
FORMAT.COM
DISKCOPY.COM
ATTRIB.EXE
...
\PROGRAM
\DBASE
DBASE.EXE
DBCACHE.EXE
INSTALL.EXE
\WORD
WORD.EXE
WORD_DCA.EXE
CAPTURE.COM
\DATA
\DB
SALES.DBF
WAGES.DBF
ACCT_NOS.DBF
\TEXT
LETTER1.TXT
PRIV.TXT
LETTER2.DOC
CONFIG.SYS
AUTOEXEC.BAT

An example of root directory

For example, suppose that you're a writer and that you write various kinds of books. You should use a different directory for each book you're writing. Within these directories you can name your files by chapter numbers (CHAPTER1, CHAPTER2, etc.).

Because the files are stored in different directories, according to the book they're associated with, each directory can contain files that have the same names. However, if you used the same directory for each book, this wouldn't be possible. Instead, you would have to think of different names for all the files and try to remember which files belonged to which project.

What are storage media?

Storage media preserve files. These are places where files can be permanently stored. On a personal computer, there are two major kinds of storage media, hard drives and diskettes.

A hard drive is a storage medium that is usually built into your computer. It contains files that you can use regularly. These files are immediately available for access. Since the hard drive is built into your computer, you cannot easily remove it to transfer data to another computer.

A diskette, on the other hand, is a storage medium that you can exchange with other users or use to transport data to another computer. You can also use diskettes to create backup copies of valuable information and programs, and to store information that is rarely used.

The term *disk drive* is closely related to the term storage medium. Different storage media are attached to certain disk drives. You address these disk drives with a letter and a colon. Disk drives for diskettes use the letters A and B, while disk drives for hard drives use the letters C and D, etc.

1.4.4 What to do if you encounter problems

The following are a few solutions to problems that often occur when working with computers.

Problem 1: Problems starting your computer

Problem

You can have problems when starting up (booting) your computer. For example, the following message could appear:

```
Non-System disk or disk error
Replace and press any key when ready
```

Solution

If you have a computer with a hard drive and you installed MS-DOS correctly, your computer will look for startup (boot) instructions on the hard drive. However, most users install MS-DOS so that the operating system is instructed to look for boot information on a diskette in drive A: first.

This is done so that during a large loss of data (for example, a hard drive crash), the computer system can be booted from a floppy diskette during an attempt to recover damaged data.

If a disk isn't found in drive A:, the computer attempts to read boot information from the root directory of drive C:. If a diskette has been placed in drive A: and this diskette doesn't contain boot information, the Non-System Disk error message will be displayed.

This problem is easily corrected. Remove the Non-System disk and press the Spacebar to continue. The computer will only boot from the hard drive if there are no diskettes in the disk drive.

If your computer doesn't have a hard drive, you must place a DOS start diskette in drive A: and pull down the drive lever. Then press the Spacebar to continue. This diskette is formatted with several files necessary for executing MS-DOS, as well as other files necessary for the operating system. Refer to the appropriate sections of your MS-DOS 5.0 users manual for more information.

Problem 2: Error message when selecting a disk drive

Problem

You are trying to select a disk drive, but the computer keeps displaying this message:

```
Not ready reading drive A
Abort, Retry, Fail?
```

Solution

You may have forgot to close the lever of the disk drive. Close it and press R (for Retry).

Problem 3: Error message when reading the diskette

You want to perform a command that requires access to a diskette, but your computer keeps displaying the message:

```
General failure reading drive A
Abort, Retry, Fail?
```

In this case, you cannot read the diskette in drive A:. Either the diskette is defective, you didn't place it in the drive correctly or it's not formatted. Press F and type the following after the message:

```
C:
```

Press Enter and you will be back in drive C:. Now you can insert the diskette correctly or try to format the diskette, if necessary.

Problem 4: No response to a command

Problem

You entered a command, but the computer does not respond.

Solution

Did you remember to press Enter? In many cases you have to press this key before the computer can execute a command.

If your computer still doesn't respond after you have pressed Enter several times, perhaps you accidentally pressed Prt Sc or PRINT. Switching on the printer could help. If your computer still doesn't respond, reboot the computer by pressing Ctrl, Alt and Del at the same time.

Problem 5: Bad command or filename

Problem

You enter a command printed in the book, but the computer continues to display the message:

```
Bad command or filename
```

Solution

It's possible that you mistyped the command. Type the command again and press Enter.

Problem 6: You accidentally enter a command

Problem

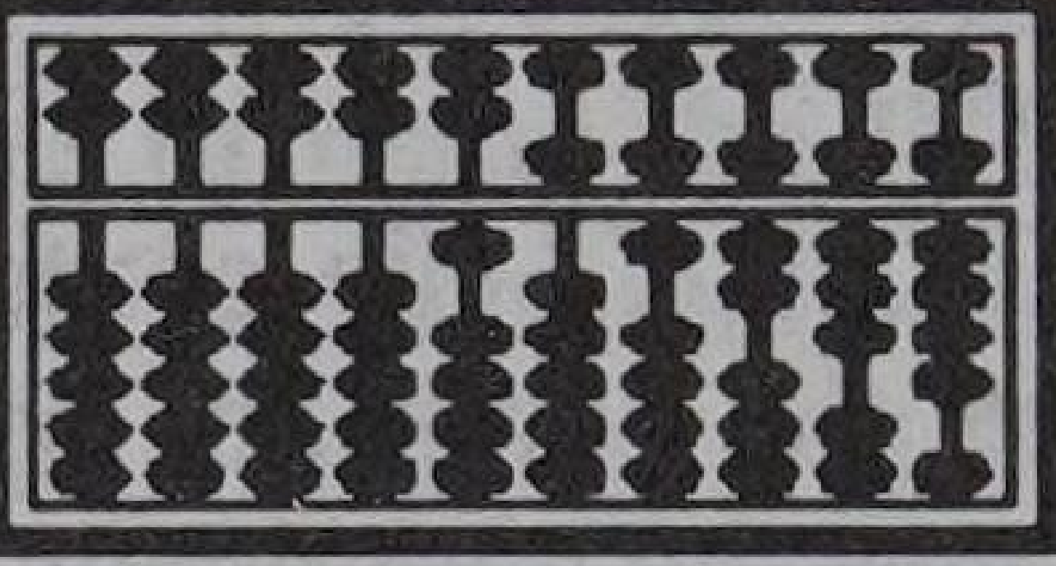
You accidentally typed FORMAT A: or a command with similar effects, but you have an important diskette in drive A: that you don't want to format. (This would delete all the data on the diskette.) You don't have any blank diskettes that you could insert in its place.

Solution

Take the diskette out of the drive and hold down the Ctrl key while pressing C. A caret with a C (^C) should appear on the screen and DOS will return with the system prompt. By doing this you aborted the FORMAT command.

This solution works with almost all DOS commands and from within many programs.

Chapter



2

Working With MS-DOS 5.0

Here we'll show you the basics of working with MS-DOS. We'll show you how to start and exit DOS, enter and work with DOS commands and start applications.

2.1 Setting Up MS-DOS

Basically, MS-DOS 5.0 is customized to your computer when you install it using the SETUP program. Installation is very easy and is controlled by menus. Most users will never run into installation problems. However, SETUP cannot handle all of the options during installation or perform any kind of fine tuning. So you cannot make all of the adjustments during installation.

Also, you might want to use some of MS-DOS' special features to optimize your system's performance or to change the current settings for later use. For example, if you change the equipment on your computer (additional memory, larger hard drive, different monitor, mouse etc.), then the operating system must be adapted to the new hardware.

You can adapt MS-DOS to the hardware (and your requirements) at any time by changing two systems files. Most of the necessary commands are stored in either the AUTOEXEC.BAT or CONFIG.SYS file.

Make changes carefully

Be very careful when making changes to either the AUTOEXEC.BAT or CONFIG.SYS files because mistakes can lead to serious problems. For example, your computer may not be able to find programs or the keyboard might not be able to produce the characters you want.

Mistakes in the CONFIG.SYS file could prevent the computer from starting (booting) successfully. If this happens the computer won't be able to operate and you may not be able to correct the mistake very easily.

You should always create a bootable backup floppy diskette before making changes to the AUTOEXEC.BAT or CONFIG.SYS files. You can then use this diskette to start your system and correct any mistakes in the files.

Setting up the AUTOEXEC.BAT file

The AUTOEXEC.BAT file contains MS-DOS commands that are automatically processed when the computer boots. This automates the starting sequence and saves time because the commands don't have to be typed at the keyboard.

An AUTOEXEC.BAT file is automatically created when you install MS-DOS. The following explanations will help you understand the contents of a typical AUTOEXEC.BAT file.

Making MS-DOS commands always available

If you installed MS-DOS in a subdirectory on your hard drive, use the PATH command to ensure that these commands can be executed at any time. For example, if MS-DOS is in the DOS subdirectory on the C: hard drive, then your AUTOEXEC.BAT should contain the following line:

```
PATH C:\DOS
```

☞ Some application programs work properly only if the directory containing the program files is included in the search path for commands. For example, Microsoft Word 5.0 is such a program. If you haven't specified a path for the WORD directory, you won't be able to use the thesaurus.

Making data files always available

Just as the PATH command lets you use commands from any directory, the APPEND command lets you access data (non-executable) files from any directory. This is important for many MS-DOS device drivers and programs that work with OVL (overlay) files. To access data files in the DOC directory on drive C: from any directory, without adding a path specification, add the following command line:

```
APPEND C:\DOC
```

Using foreign character sets

The KEYB command is used to adapt MS-DOS to foreign character sets. For example, to change to the U.K. keyboard, which will replace the # sign with the £ sign, the AUTOEXEC.BAT file should contain the following line:

```
KEYB UK,437,C:\DOS\KEYBOARD.SYS
```

☞ In some cases, the KEYB command is already installed in the CONFIG.SYS with the MS-DOS 5.0 SETUP command. Under these circumstances you don't necessarily need the line in the AUTOEXEC.BAT.

You should be aware of the advantages of using the INSTALL command in the CONFIG.SYS file. Using the INSTALL command in the CONFIG.SYS file allows you to load memory resident programs into memory before the AUTOEXEC.BAT file is processed. The CONFIG.SYS is processed only once at the beginning of the boot process. But the AUTOEXEC.BAT file can be processed (even accidentally) more than once. However, running the AUTOEXEC.BAT file more than once may result in error messages with certain MS-DOS commands (PRINT /D or FASTOPEN /X). If you load these commands using INSTALL from the CONFIG.SYS file, you won't accidently try to run these command again while the computer is running.

Specify the location of the command interpreter

When certain application programs are started, they overwrite part of the COMMAND.COM program in memory. Therefore when the application program ends COMMAND.COM must be reloaded. So that MS-DOS knows where on the hard drive to find the command interpreter is the following in the AUTOEXEC.BAT:

```
SET COMSPEC=C:\COMMAND.COM
```

☞ Enter the complete path for the COMMAND.COM.

For example, if COMMAND.COM is in the DOS subdirectory of your hard drive, the following command line should be in the AUTOEXEC.BAT file:

```
SET COMSPEC=C:\DOS\COMMAND.COM
```

You also need to insert the SHELL command in the CONFIG.SYS.

Set the system prompt

Use the PROMPT command to define the appearance of the MS-DOS system or ready prompt. It's a good idea to specify a system prompt that displays the current directory in the AUTOEXEC.BAT:

```
PROMPT $P$G
```

Set the break option

Use the MS-DOS BREAK command to determine when the Ctrl+C key combination for aborting a command can be used. Add the following line to your AUTOEXEC.BAT file to switch on the break option when accessing a diskette or hard drive:

```
BREAK ON
```

This option is especially important when using application programs and commands that require intensive diskette or hard drive access.

Allow screen printouts in graphics mode

When a color or graphics card adapter is used, the MS-DOS GRAPHICS command enables you to print a graphics display screen. Add the following line to your AUTOEXEC.BAT file:

```
GRAPHICS
```

Display special characters in graphics mode

In order to use the expanded ASCII characters 128 to 255 in graphics mode, use the GRAFTABL command to load a table of these characters in the memory. This command increases the size of MS-DOS that is resident in memory. It also enables you to use character sets for different countries.

```
GRAFTABL
```

Set a character set table

After preparing all of the devices to be used with character set tables (character sets are stored in Codepages), you can activate the new character set with CHCP. CHCP stands for CHange CodePage. The following command lines adapt the screen of an EGA card and the keyboard to the multilingual international table:

```
MODE CON CP PREP=((850) C:\DOS\EGA.CPI)
KEYB UK,850,C:\DOS\KEYBOARD.SYS
CHCP 850
```

☞ You only need to install a character set if you require special characters.

Set a temporary directory

Certain MS-DOS commands create temporary files. For example, this happens when more than one command is combined with the pipe symbol "|". Older versions of MS-DOS would place these temporary files in the root directory or in the current directory, deleting them at the end of the command line.

The MS-DOS Shell creates a scratch file named DOSCMD.BAT. Since these temporary files can be annoying, a temporary directory can be set up. To do this, you first set up an environment variable named TEMP, which supplies the directory name. For example, if you have a directory named C:\DOSTEMP and want to use it as a temporary directory, your AUTOEXEC.BAT should contain the following command line:

```
SET TEMP=C:\DOSTEMP
```

Unlike MS-DOS, many other programs (such as Microsoft Word 5.0) use TMP instead of TEMP to set a temporary working directory. For these types of temporary files, you would define the directory as follows:

```
SET TMP=C:\DOSTEMP
```

Increase the amount of conventional memory

If your computer uses an 80386 or 80486 microprocessor and has more than 640K of memory, the amount of conventional memory available for application programs may be increased by loading memory resident programs into additional memory. Most systems have 384K of additional memory, which is usually set aside for the operating system.

Some portions of this additional memory, called upper memory blocks (UMB), are left unused. Usually these UMB's aren't available for running device drivers or applications. However, if you run the EMM386 memory manager, you can access these UMB's.

In order to load device drivers and other programs into UMB's, the DEVICEHIGH or LOADHIGH command must be used to install these drivers and programs from the AUTOEXEC.BAT file. To load the code pages for the United Kingdom keyboard driver into an upper memory block of reserved memory instead of conventional memory, use the following command line:

```
LOADHIGH KEYB UK,,C:\DOS\KEYBOARD.SYS
```

Example

A simple AUTOEXEC.BAT for MS-DOS in the DOS subdirectory of the C: hard drive might look as follows:

```
PATH C:\DOS
C:\DOS\KEYB UK,437,C:\DOS\KEYBOARD.SYS
SET COMSPEC=C:\DOS\COMMAND.COM
BREAK ON
PROMPT $P$G
SET TEMP=C:\TMP
```

More Hints

If your computer doesn't have a battery operated clock, include the following lines in your AUTOEXEC.BAT:

```
DATE
TIME
```

Your computer will prompt you to enter the date and the time whenever you start or reboot your system. This will ensure that the date and time stamp for your files are always current.

In order to print files from the MS-DOS Shell, you must use the MS-DOS PRINT command with the /D switch before calling the MS-DOS Shell. Otherwise the menu command for printing won't print. The /D switch defines a device for the print queue for printing files. Use the following command line in your AUTOEXEC.BAT:

```
PRINT /D:LPT1:
```

This initializes the print queue through the first parallel port. However, this uses a portion of your computer's conventional memory area. So you should only use this option when you need to print in the MS-DOS Shell.

Load the MS-DOS Shell automatically when you start the computer

To install and run the MS-DOS Shell from the AUTOEXEC.BAT, simply use the DOSSHELL command. This command should be the last command in the AUTOEXEC.BAT file. Otherwise, any commands which follow may not be activated.

Setting up the CONFIG.SYS file

The following commands and options apply only to the CONFIG.SYS file. In the following examples, we assume that MS-DOS is installed in the C:\DOS directory of the hard drive. If this isn't the name of your DOS directory, change the examples to the actual name of your DOS directory.

The explanations are intended to help you understand the commands in the CONFIG.SYS. When you install MS-DOS, the CONFIG.SYS file is created automatically and the installation procedure adapts it to your computer.

Adjust the date and currency format

Use the COUNTRY command to adapt MS-DOS to the date and currency format of a country. For example, use the following setting for the United Kingdom:

```
COUNTRY=044,437,C:\DOS\COUNTRY.SYS
```

The computer will display the date and time in the format used in the United Kingdom.

Optimize access to disk drives

The BUFFERS statement can be used to speed up access to diskettes and hard drives. With a system that has 640K conventional memory and a hard drive, type the following in the CONFIG.SYS file:

```
BUFFERS=20
```

If you use the special driver program SMARTDRV.SYS to speed up access, use the setting BUFFERS=30 for BUFFERS. Higher settings not only waste valuable memory, but could actually reduce the speed of your computer.

Set the number of files opened simultaneously

Many application programs may require more than the 8 files that are provided by the default settings for MS-DOS. This number can be increased with the FILES command. For example, you could include the following line in your CONFIG.SYS file:

```
FILES=20
```

☞ You may need even more files for some database programs. Check the program's manual or increase the value if an error message appears.

Set the number of logical disk drives

By default, MS-DOS can manage 5 disk drives lettered A-E. To define more disk drives, use LASTDRIVE. You might use this command if your hard drive has multiple partitions or you redefine disk drives. To use 8 disk drives lettered A to H, type the following command line in the AUTOEXEC.BAT:

```
LASTDRIVE=H
```

Define a command interpreter

Usually MS-DOS loads the COMMAND.COM program from the root directory of the current disk drive as a command interpreter. You specify an alternate path or filename with SHELL. If the command interpreter is in the DOS subdirectory, use the following line:

```
SHELL=C:\DOS\COMMAND.COM /P
```

☞ The path and the name of the command interpreter you specify must be the same as those specified by COMSPEC in the AUTOEXEC.BAT file. Make sure you specify the /P switch. This switch makes

COMMAND.COM permanent. You can't quit the command interpreter by entering Exit.

Determine the size of the system environment

MS-DOS stores important information in an area called the system environment. You can increase the amount of memory reserved for the environment variables with the SHELL command. Since 512 bytes of system environment is enough for most applications, change the size in the CONFIG.SYS file as follows:

```
SHELL=C:\DOS\COMMAND.COM /P /E:512
```

☞ Be sure that you specify the /P switch; otherwise when you re-start the computer it won't run the AUTOEXEC.BAT system file. Specifying this switch also guarantees that the command interpreter is permanently installed in memory and that you can't exit from the command interpreter.

Allow for additional functions for your screen and keyboard

A special driver program, called ANSI.SYS, can be used to change the keyboard and the type of screen output. ANSI.SYS is loaded by using the DEVICE command. Install ANSI.SYS by using the following line in CONFIG.SYS:

```
DEVICE=C:\DOS\ANSI.SYS
```

Accelerate access to files

FASTOPEN speeds up the access of files that are used often. This is especially useful for database programs. Insert the following command line:

```
INSTALL=C:\DOS\FASTOPEN.EXE C:=100
```

Use additional memory

MS-DOS can directly manage only 640K of memory as conventional memory. If your computer has more memory, certain application programs can use the additional memory as Expanded Memory. However, you'll need a special auxiliary program (Memory Driver) that is not part of MS-DOS.

MS-DOS can use extended memory (Extended Memory) as a RAM disk. Use the RAMDRIVE.SYS device driver (refer to, "Use memory as an additional disk drive"). You can also use extended memory to speed up access to your hard drive by activating the SMARTDRV cache program from the CONFIG.SYS file. SMARTDRV is also a device driver.

☞ Use the MEM command to see if your computer has extended memory, The information that's displayed may look similar to the following:

```
 655360 bytes total conventional memory
 655360 bytes available to MS-DOS
 564160 largest executable program size

1441792 bytes total contiguous extended memory
      0 bytes available contiguous extended memory
 589824 bytes available XMS memory
        MS-DOS resident in High Memory Area
```

Total memory refers to the amount of conventional memory that MS-DOS can directly use, while the "largest executable program size" indicates how much memory is available when working with commands and programs. We'll discuss extended memory in more detail in the following section.

Special memory features

Installing and using more than 640K isn't always easy. Usually it requires a knowledge of memory management. Until now, distinguishing between conventional memory, extended memory and expanded memory was very complicated, which obviously made it difficult to use.

Although total memory management is more efficient and safer with MS-DOS 5.0, it's not easy to achieve an "optimum" installation of extended memory.

The following features can be used with the CONFIG.SYS:

- Before MS-DOS 5.0, extended memory was used mainly through the system call 15 (INT 15). However, this often caused multiple programs to collide, resulting in a system crash. Now a special driver program called HIMEM.SYS is used to manage extended memory.

- HIMEM.SYS also prepares a special memory area called High Memory. Under certain circumstances, MS-DOS can use this memory, thus increasing the amount of available conventional memory. To do this, you must activate HIMEM with the DEVICE command in CONFIG.SYS. You must also use the DOS=HIGH switch in the CONFIG.SYS.

- You can also load driver programs (RAM disk, SMARTDRV, mouse drivers etc.) in a special memory area (Upper Memory) to increase the conventional memory available for programs. Use the UMB setting after the DOS= switch. To use both High Memory and Upper Memory in MS-DOS, include the following command line in CONFIG.SYS:

  ```
  DOS=HIGH,UMB
  ```

Use memory as an additional disk drive

You can speed up access to diskettes by using part of the conventional memory as a RAM drive instead of a mechanical disk drive. Use the RAMDRIVE.SYS device driver to do this.

Insert this line in the CONFIG.SYS file:

```
DEVICE=C:\DOS\RAMDRIVE.SYS
```

✎ Unlike the SMARTDRV program for temporary storage of hard drive contents, the RAMDRIVE program uses extended memory. So you must reserve memory using HIMEM.

After defining a RAM drive, the next time you boot your computer, a new disk drive will be available. This drive will have a capacity of 64K and can contain 64 files or directories. The computer displays an informational message when you start your computer. Unfortunately, since the memory for the RAM disk comes from conventional memory, memory hungry applications may not have enough memory to run.

☞ In PC-DOS the driver is called VDISK.SYS instead of RAMDRIVE.SYS.

If your computer has extended memory, you can use it for the RAM disk. This takes some of the burden off of conventional memory. Insert the /E switch in the command line. Many ATs with 1 Meg of total memory have 384K of extended memory. To use this extended memory for a RAM disk, include the following command line in CONFIG.SYS instead of the one above:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 384 /E
```

Expanded memory can also be used for a RAM disk. Use /A as a switch. If you have 1024K of expanded memory, you can use it as an additional disk drive with the following command line:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 1024 /A
```

Accelerate access to hard drives

If your computer has enough extended or expanded memory, you can speed up access to your hard drive by using the SMARTDRV.SYS cache program. The parameters following the SMARTDRV command specify how much extended memory to use for temporary storage. If an amount isn't specified, a value of 256K is used. In order to install 1024K of extended memory for temporarily storing data, type the following command line in the CONFIG.SYS:

```
DEVICE=SMARTDRV.SYS 1024
```

Extended memory uses the MS-DOS 5.0 SMARTDRV program which uses an XMS driver. So, before using SMARTDRV you must install the HIMEM.SYS driver.

If you use available expanded memory instead of extended memory, use the /A switch. The line above would then read:

```
DEVICE=SMARTDRV.SYS 1024 /A
```

Define additional disk drives

MS-DOS recognizes all disk drives that are properly installed. To change the defined size of a disk drive, use the DRIVER.SYS device driver.

For example, if you have a 3.5 inch 1.44 Meg drive as drive B:, you can define this drive to also act like a 720K capacity drive rather than a 1.44 Meg capacity drive, change CONFIG.SYS to include the following line:

```
DEVICE=C:\DOS\DRIVER.SYS /D:1 /F:2 /H:2 /S:9 /T:80
```

The next time your computer is warm started, MS-DOS will report an additional loaded driver, indicating the disk drive letter to specify when you want to use disk drive B: as a 720K disk drive. Normally disk drive B: will remain a 1.44 Meg disk drive named B:. When you access the new drive, it will act like a 720K drive.

If the DRIVER.SYS file isn't in the root directory, specify the complete path.

Example

A sample CONFIG.SYS file for MS-DOS located in the DOS subdirectory of the C: (hard) drive should look similar to the following:

```
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH,UMB
DEVICE=C:\DOS\ANSI.SYS
COUNTRY=044,,C:\DOS\COUNTRY.SYS
FILES=20
BUFFERS=20
SHELL=C:\COMMAND.COM /P /E:500
INSTALL=C:\DOS\FASTOPEN.EXE C:=100
```

Use Upper Memory

If your computer has extended memory available as Upper Memory, you can also load driver programs (extension .SYS) into this special memory area. This provides more conventional memory for application programs. You can use the DEVICEHIGH command in place of DEVICE:

```
DEVICEHIGH=C:\DOS\ANSI.SYS
```

If your computer doesn't have Upper Memory or there isn’t enough room, the computer automatically loads the driver in the conventional area (Conventional memory up to 640K).

2.2 Starting MS-DOS

Although you can run MS-DOS 5.0 from a floppy diskette after installing it, you might have problems with some commands. So, use the SETUP installation program and install MS-DOS so that it can be adapted to your computer. If you haven't installed MS-DOS 5.0 yet, do so now.

The methods for starting MS-DOS vary depending on your computer and the kind of installation.

☞ If your computer is switched on and you want to re-start it, you don't have to switch off the computer. Simply press Ctrl + Alt + Del (warm start). This so-called warm start saves wear and tear on the power supply and monitor of your computer.

Starting MS-DOS from a diskette

You'll need a system diskette to boot your computer. This diskette must have been previously formatted with the /S switch of the FORMAT command and will contain the MS-DOS files and commands. For example, the diskette should include the AUTOEXEC.BAT and CONFIG.SYS files we discussed in the previous section.

Insert the system diskette

Insert the system diskette into drive A: and close the disk drive door (if you have one).

☞ If you're not sure which of your disk drives is labeled A, don't insert a diskette. Instead, see which disk drive light goes on first. Usually this is drive A:. Otherwise, refer to your computer manual.

Switching on your computer

Switch on your monitor, then switch on the computer. The screen will display various messages, depending on the kind of computer you have.

If your computer doesn't have a battery operated clock, you're prompted to enter the date and the time. The screen displays the default values and the notation for making the entries. Enter the date and the time (don't forget to press Enter after each entry). If you press Enter without making entries the default values for the date and the time will remain.

Computer ready for operation

Your computer is ready for operation and you can begin entering commands when the system prompt appears on the screen:

```
A:\>_
```

Even if MS-DOS is installed on your hard drive, you should still make a system diskette and keep it available. Sometimes it may be impossible to start MS-DOS from your hard drive, which would obviously prevent you from using your computer. For example, this could happen if mistakes were made while changing the CONFIG.SYS file. Your computer would fail when you try to start it.

Special features about entering commands

When booting your computer from a system diskette, you could have some problems working with your computer. This usually occurs if your hard drive contains all of the MS-DOS commands. The commands exist, but the computer can't find them.

If you start MS-DOS from a diskette and the commands are on the hard drive, define a search path on the directory with the commands. For example, if the DOS directory on the hard drive is C:\DOS, use the following line to create your search path:

```
PATH C:\DOS
```

Instead of defining a search path, you can type the complete path in front of any MS-DOS command. For example, to format a diskette in drive A: type the command line:

```
C:\DOS\FORMAT A:
```

If your computer doesn't have a hard drive, you'll have problems working with MS-DOS because all of the MS-DOS commands do not fit on a single diskette. Instead, they are stored on several diskettes.

So copy the commands you use most often to your system diskette. Then copy the remaining commands to other diskettes. If the message "Bad command or filename" appears and you typed the command correctly, insert the diskette containing the command.

Starting MS-DOS from the hard drive

Fortunately, if you have a hard drive, you won't need a diskette to start the computer. However, remember to remove any diskettes from drive A: before attempting to boot your computer. MS-DOS will only start from the hard drive if a diskette isn't in disk drive A:

Switch on your computer

Always switch on the monitor first before switching on your computer. As your computer starts you'll see different messages on the screen, depending on the kind of computer you have.

Enter the date and the time

If your computer doesn't have a battery operated clock, you'll be prompted to enter the date and the time. The screen displays the preset values and notation for entering the date and time. Enter the date and the time, pressing Enter after each entry. If you press Enter without making any entries, the default values will remain.

Computer ready for operation

Your computer is ready for operation when the system prompt appears on the screen followed by the cursor:

```
C:\>_
```

Now you can enter commands. If you've set up your system to start the MS-DOS Shell to start automatically, instead of seeing the system prompt, you'll be in the MS-DOS Shell. You can either start the command interpreter from the MS-DOS Shell with the "Command Prompt" program item or exit the MS-DOS Shell and remove it from memory by pressing F3.

Quitting

If you aren't running an application program and the computer is displaying the system prompt, your computer may be switched off at any time. However, remove all diskettes from the disk drives before switching off your computer.

☞ Before switching off your computer, you should quit the application programs properly. Turning off the power to your computer while running an application program could cause loss of valuable data. You should also secure the heads of your hard drive before switching off the computer. Many hard drive manufacturers supply a PARK program with their software. This program, which isn't part of MS-DOS, moves the heads of the hard drive to a shock-protected position.

If you're in the MS-DOS Shell, before switching off your computer exit the MS-DOS Shell by pressing F3 and secure your hard drive by using PARK. This is especially important if you've started programs, from the MS-DOS Shell, that you still haven't quit. The MS-DOS Shell stores information in temporary files. If you don't exit from the MS-DOS Shell properly, these temporary files won't be deleted correctly when you've completed your work.

2.3 Using The Keyboard

The keyboard of a computer is similar to a typewriter keyboard, except for some additional keys which are used for computer related functions.

Additional computer keys

Ctrl
Alt
Del
Esc
Pause (also referred to as <Break>)
Num Lock
Caps Lock
Ins
Home
End
PgDn
PgUp
Prt Sc
Scroll Lock

Special functions of keys in MS-DOS

Enter	After typing a command, pressing this key causes the computer to execute the command. It's usually larger than the letter keys and located on the right side of the keyboard.
Spacebar	This is the wide key at the bottom of the keyboard.
Backspace	Correction key, also called the Backspace key. This key is usually found above the Enter key with an arrow pointing to the left. You can use this key to delete the character before the cursor. Continue pressing the Backspace key to delete additional characters until there are no characters in front of the cursor.
← → ↑ ↓	Although these cursor keys can move the cursor anywhere on the screen in application programs, they don't have many functions within MS-DOS. However, MS-DOS 5.0 includes a special auxiliary program called DOSKEY to simplify entering and changing command lines, especially when working with frequently used DOS commands.
Ctrl	Special key for producing control key combinations.

Control key combinations

Ctrl+S	Pauses the output of data to the screen. Press any key or Ctrl + S again to continue displaying data on the screen.
Ctrl+C	Interrupts a command and returns to the system prompt.
Ctrl+Alt+Del	Re-starts your computer (executes a warm boot).

☞ When you execute a warm boot, all of the data in the conventional memory area of the computer is lost.

Ctrl Prt Sc	Switches output to the printer on or off.

Keyboard shortcuts for the command line

To simplify your work at the command line, you can use one of the special function keys or editing keys. Additional keys are available to edit a command line if you use DOSKEY. These functions are discussed in detail in Section 2.5.

F1	Pressing F1 copies the template of the previous command, one character at a time, to the current cursor position. Pressing F1 repeatedly will copy the entire template.
F2	Retypes characters from the template up to a matching character. The matching character is specified by pressing its key after pressing F2.
F3	Copies any remaining characters in the template. For example, typing DIR *.BAK and pressing Enter will display all files with the .BAK extension in the current directory. Typing DEL and pressing F3 will type the remainder of the command into the command line: DEL *.BAK. Pressing F3 when the cursor is at the beginning of a new command line will retype the complete template.
F4	Skips over characters in the template up to a matching character, without copying the characters to the command line. The matching character is specified by pressing its key after pressing F4.
Del	Erases the character at the cursor position and moves forward one character in the template.
Ins	Switches between the insert and overstrike modes. In the insert mode, new characters are inserted without typing over the previously entered characters. Move the cursor with the left and right cursor control keys.
Backspace	Erases the character in front of the cursor on the command line and moves forward one position in the template.

2.4 Entering And Repeating Command Lines

When you type at the system prompt, MS-DOS saves a copy of the last line that you typed in a temporary location called a *template*. To save you the trouble of retyping long command lines, you can redisplay (recall) this command line, change it if you wish and execute it a second time.

Entering and correcting new command lines

A command line is entered after the system prompt. You can type a command in either uppercase or lowercase - commands are not case sensitive. The Backspace key can be used to delete the character to the left of the cursor so that you can correct a typing error. MS-DOS copies the command line to a temporary storage location, called the template, and executes it as soon as you press Enter. The template saves only the last command line entered.

Repeating a command line

If the cursor is located at the start of a command line, press F3 to display the previous command. The previous command is copied from the template to the current command line and the cursor is placed at the end of the command line. Press Enter to execute the command again.

The following examples are techniques for editing the command line.

Example 1

Suppose that you want to display all of the files in the DOS directory that have the .EXE extension. Instead of typing:

```
DIR C:\DOS\*.EXE
```

you accidentally type the following command line:

```
DIR C:\DOS\*.EYE
```

Press the Backspace key twice to delete the last two characters. Then enter the correct characters (XE) and execute the command line by pressing Enter.

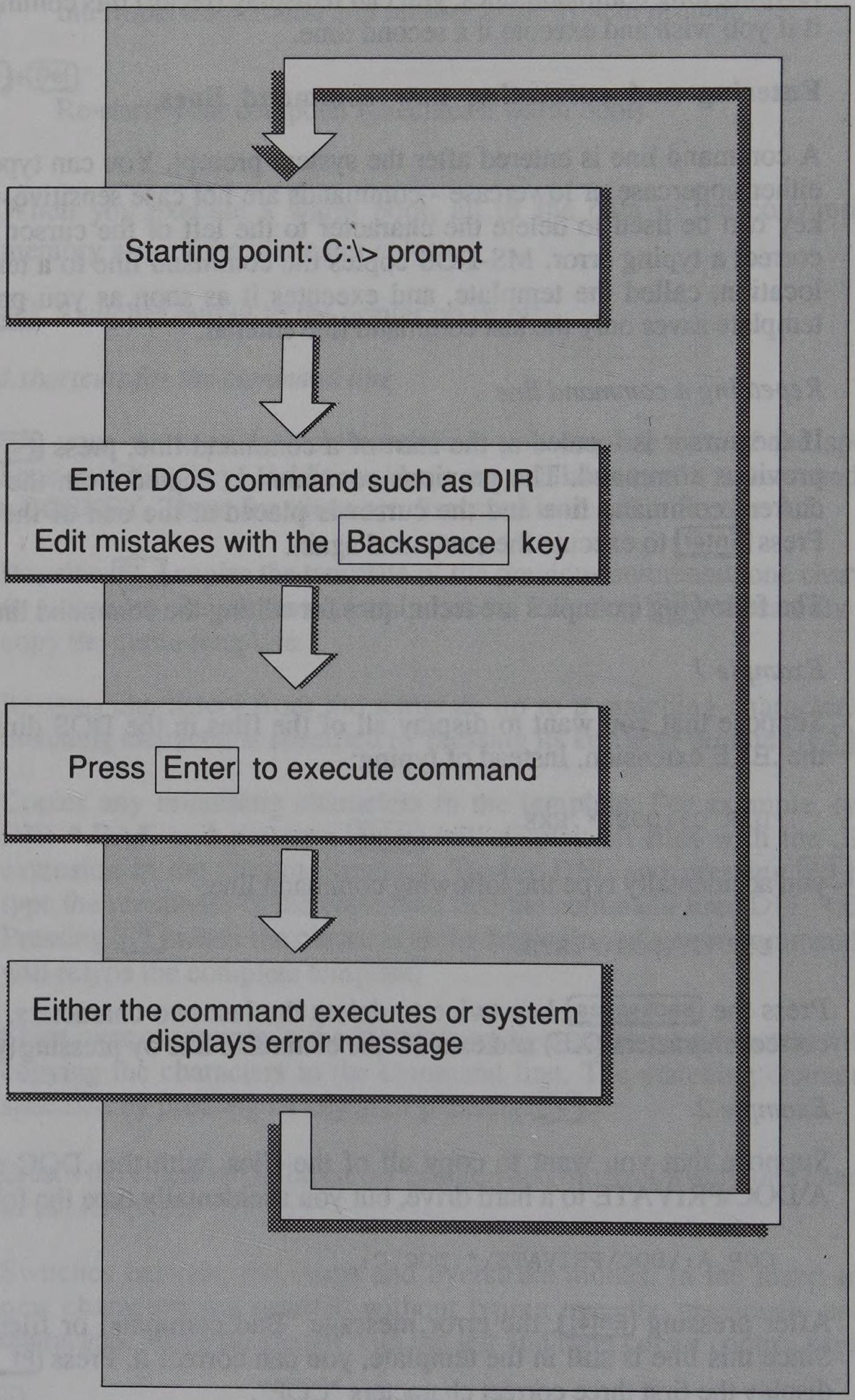
Example 2

Suppose that you want to copy all of the files, with the .DOC extension, from A\DOC\PRIVATE to a hard drive, but you accidentally type the following line:

```
COP A:\DOC\PRIVATE\*.DOC C:
```

After pressing Enter, the error message "Bad command or filename" appears. Since this line is still in the template, you can correct it. Press F1 three times to display the first three correct characters "COP".

Then press Ins to change to insert mode. Type Y to insert the correction at the current position and then copy the rest of the line from the template to the current command line by pressing F3. Press Enter to execute the correct command line.


Starting point: C:\> prompt
Enter DOS command such as DIR
Edit mistakes with the Backspace key
Press Enter to execute command
Either the command executes or system displays error message

Entering and changing commands at the system prompt (C:\>)

2.5 Entering And Correcting Commands With DOSKEY

As many users know, it can become very time-consuming to enter and correct commands in MS-DOS. However, MS-DOS 5.0 has a utility command called DOSKEY, which simplifies using the DOS command line. This program is loaded into the computer's memory when you start your computer. DOSKEY also lets you to create batch files and macros easily and can even replace external and internal MS-DOS commands with modified versions.

Running DOSKEY

DOSKEY is a TSR-Program (**T**erminate and **S**tay **R**esident Program), which means that after being used the first time, the program remains in memory until the computer is restarted. This command can be entered without any parameters or other specifications. After calling it from the command line or from an AUTOEXEC.BAT file, your computer will load DOSKEY and display the message "DOSKey installed". Although this program uses approximately 4K of memory, its versatility and usefulness are worth it.

DOSKEY lets you use the cursor keys to recall, edit and/or re-execute commands that have been stored in a special 1024 character "history" buffer. (The size of the history buffer may be changed, which we'll explain later.) Several parameters and switches can be used to control the way DOSKEY works.

Insert or overstrike mode

DOSKEY simplifies editing the command line. After it is installed, you can use the ← and → cursor keys to move the cursor or press the Del key to erase characters. Depending on the input mode, either insert new characters at the current cursor position or type over existing characters. These are the switches for setting insert or overstrike modes:

/INSERT	As its default setting, DOSKEY automatically uses insert mode to edit commands. Press the Ins key to switch between insert and overstrike mode.
/OVERSTRIKE	Installs overstrike mode when DOSKEY is installed in memory. So characters entered in this mode replace any existing characters at the current cursor position.

As with other command switches, you specify the switch after typing the command name (e.g. DOSKEY /OVERSTRIKE).

Changing the size of the line buffer

DOSKEY stores command lines in a special history buffer, which by default is 1024 characters in length. When this memory area is full, the oldest lines are removed to make room for new lines. In most cases, you'll find that a 1024 character history buffer is sufficient for storing command lines. However, macros are also stored in this buffer. So if you use macros, consider increasing the size of the buffer.

/BUFSIZE=	Specifies the size of the buffer for command lines and macros. By default DOSKEY reserves 1024 bytes of memory for this. Once DOSKEY is installed you can only change this parameter with /REINSTALL.
/REINSTALL	Installs another copy of DOSKEY in conventional memory. This is needed, for example, to change the size of the history buffer with /BUFSIZE. However, for every new copy you lose about 4K of memory.

Displaying the history buffer lines and macros

Command lines and macros stored in the DOSKEY history buffer can be displayed on the screen. By re-directing this information, you can also print lists to the printer or create batch files.

/MACROS	Displays a list of all of DOSKEY macros that are currently defined. The macros are displayed in the format Name=Commands. By using the redirection symbol ">", the macro list can be sent to the printer or a file.
/HISTORY	Displays a list of all of the commands stored in the DOSKEY history buffer. By using the redirection symbol ">", the macro list may be sent to the printer or a file. Also, by redirecting this list to a file and appending the .BAT extension, a batch file may be created.

You can also display the history buffers contents by pressing F7. DOSKEY displays each command line with a line number. By pressing the F9 key and entering a line number you can recall a specific line for editing or re-execution.

Editing the current command line

As we mentioned earlier, the command lines are stored in the history buffer. Pressing the ↑ and ↓ cursor keys will scroll the stored command lines.

The current line can be edited by using the ← and → cursor keys to move the cursor within the line. Use the Del key to erase characters. If you enter the wrong character, simply type over it with the correct character (with overstrike mode switched on). To insert characters that are missing, switch to insert mode (if it isn't already on) by pressing Ins and insert the desired character in the current cursor position.

You can also start DOSKEY directly in insert mode by using the /INSERT switch. Notice how the cursor changes appearance when you switch to insert mode. Press Ins again to switch the cursor back to normal.

To move the cursor to the beginning or end of a command line, use the Home and End keys respectively. The cursor can be moved from word to word by holding down the Ctrl key while pressing the ← or → cursor keys. Pressing Ctrl + End deletes all of the characters from the current cursor position to the end of the line. Ctrl + Home can be used to delete all of the characters from the current cursor position to the beginning of the line. The rest of the line remains unchanged.

Repeating a command line

To display the preceding line in the history buffer, press the ↑ key. If you've just started DOSKEY and haven't entered any commands, nothing will happen because DOSKEY hasn't stored any lines yet.

After you call DOSKEY you can use the editing options of the function keys. For example, press F3 to edit the last entered command line. Other special keys are also defined; these are discussed later in this chapter.

PgUp displays the first stored line and PgDn displays the last stored line in the history buffer.

Search and display options for command lines

If you lose track of the contents of the history buffer, press F7 for a complete list of all of the lines. The lines in the list will be numbered. An example may look similar to the following:

```
1: dir
2: cd dos
3: dir *.exe
4: cd..
```

If more than 24 lines are in the command buffer, the following message appears:

```
-- More --
```

To continue the display, press any key.

You can then edit or execute a specific line without having to repeatedly press the ↑ or ↓ cursor keys. To do this press the F9 key. The screen displays the prompt "Line number". Then, type the line number of the desired command.

In the above example, to execute the command line "DIR *.EXE" again, first press F9, then enter three for the third line and press Enter. The line appears as the current command line. Edit or execute it again by pressing Enter.

The following is a method for retrieving a command line that was already entered. Simply type the beginning of the line. For example, type the following command lines and press Enter after each line.

```
...
COPY C:\DOS\*.COM A:
DIR A:
CHKDSK A:
```

Suppose that you want to copy the COM files, which were just copied onto a diskette in drive A:, to another diskette. This means that you'll be using the command line "COPY..." a second time. Simply enter a part of the line and press F8. For example, type CO and press F8. DOSKEY searches for the first appropriate line and displays it on the command line for editing or execution. For our example, simply type:

```
COF8
```

If the command you wanted wasn't displayed, you may have used COPY to copy a different file earlier. Simply press F8 again. DOSKEY will then search for the next appropriate line.

Creating batch files automatically

A batch file is a text file containing a series of MS-DOS commands. They are often used to simplify repetitive or complicated tasks. When you type the name of a batch file at the system prompt, MS-DOS executes the lines stored, just as if you typed them at the keyboard - except that you didn't have to type them.

For many DOS users, batch files are a mystery. Although they are aware of batch files, they're hesitant to use them because they equate creating batch files with "programming". So they think working with batch files is too complicated.

DOSKEY makes it easy to create batch files. DOSKEY stores lines in its buffer similar to a batch file. However, you cannot run these lines one after the other except when pressing the ↑ and ↓ cursor keys to recall the appropriate commands. The following is a simple procedure for creating a batch file from commands that are directly entered at the keyboard:

1. Press Alt + F7 to clear the contents of the history buffer.
2. Then enter the MS-DOS command lines in the usual way. DOSKEY records them in the command buffer as DOS executes the commands.
3. To create a batch file called WORK.BAT from these executed command lines, type the following command line:

```
DOSKEY /HISTORY >WORK.BAT
```

This causes DOSKEY to write all of the lines, without line numbers, to a file named WORK.BAT. Since you're redirecting the contents of the history buffer to the WORK.BAT file, your batch file is almost complete. However, there is one little catch. DOSKEY also recorded the last command (the command line for creating the batch file: "DOSKEY /HISTORY >WORK.BAT") and included it in the WORK.BAT file. This command must be removed with an editor.

You can use EDIT, the MS-DOS editor to "fix" this batch file. To fix WORK.BAT type:

```
EDIT WORK.BAT
```

Now delete the last line ("DOSKEY /HISTORY >WORK.BAT") and save the finished batch file.

☞ If you discover that you'd like to create a batch file from the history buffer but didn't first clear the contents by pressing Alt + F7, then you can still use this method. Type:

```
DOSKEY /HISTORY >FILE.BAT
```

Edit the file FILE.BAT, delete any unnecessary command lines and save the results.

Working with macros

Besides saving commands in the history buffer and providing a flexible way to recall, edit and execute DOS commands, DOSKEY also lets you create and execute macros. A macro is like a simple batch file. Creating a macro is giving a name to a series of commands. To do this, use a notation similar to the one for defining environment variables using SET. First, enter the name of the macro followed by an equal sign. Following the equal sign, enter the series of commands that are to be performed. DOSKEY can distinguish a macro definition from a command by the = sign. Type the command line as follows:

```
DOSKEY NAME=Commands
```

Separate commands by using the character sequence $T. Parameters can be passed in a macro, as with batch files. Instead of using %1 to %9, use $1 to $9 as parameters specified from the command line.

$* stands for the entire parameter specification, including all special characters (spaces etc.) entered on the command line. If you enter more than one parameter, you can address all of the parameters at once with $*.

Suppose that you want to sort a directory by filenames, but you don't want to enter the parameters /O:n every time you want to use this command. Simply define a macro named DIRS with the appropriate command line:

```
DOSKEY DIRS=DIR /O:n
```

Now, when you need a sorted directory, simply use the new "command" DIRS. By doing this you've created a new MS-DOS macro.

The following is an example of how you could define a macro named DISK for formatting and checking a diskette:

```
DOSKEY DISK=FORMAT A: $T CHKDSK A:
```

Passing parameters in macros

Similar to batch files, you can pass parameters to DOSKEY macros by using $1, $2, etc. Suppose that we want to expand our example, DISK, so that either drive A: or B: can be selected. To do this, we must replace A: with a parameter as shown below.

```
DOSKEY DISK=FORMAT $1 $T CHKDSK $1
```

Now you can start the macro using drive A: by specifying the disk drive identifier as a parameter:

```
DISK A:
```

You can use more than one parameter by using $2, $3, etc. The parameter $* contains a complete list of parameter specifications. When you specify three parameters after a macro name, they are available separately as $1, $2 and $3. However, all three, including the spaces in between, are in the parameter $*.

Replacing internal MS-DOS commands

You can also replace internal MS-DOS commands with DOSKEY macros. DOSKEY always checks its list of macro names before executing a DOS command. If it finds a matching name in its list, DOSKEY executes the macro. If it doesn't find a matching name, DOSKEY passes the command onto the DOS command interpreter which will then execute the command.

As a test, define a macro named DIR and give it an entirely different function, as below:

```
DOSKEY DIR=ECHO DIR is forbidden...
```

Then enter the MS-DOS DIR command from the command line. Instead of seeing the current directory on the screen, you'll see the message defined in our macro.

In this way, MS-DOS 5.0 can be used to adapt normal MS-DOS commands to your own needs. Prior to MS-DOS 4.0, this could only be accomplished by using batch files.

If you wanted DIR to always display files by page, you can define a macro as follows:

```
DOSKEY DIR=DIR /P $1 $2 $3
```

When DIR is called without any other parameters, DOSKEY substitutes the character sequence DIR /P. If another parameter is specified after DIR, $1 is replaced by this parameter. DOSKEY changes "DIR C:\DOS" to "DIR /P C:\DOS". You can specify up to three parameters in this definition of the DIR macro.

If you define a "replacement" macro, you can still execute the "real" DOS command by beginning the command line with a space. In this case, DOSKEY does not substitute the macro, so the true DOS command is executed.

Using special characters in macros

Certain characters have a special meaning in command lines. These characters are immediately analyzed by the command interpreter and understood as commands. The two characters "<" and ">" for redirecting input and output as well as the pipe symbol "|" for linking commands are among these special characters. If you entered these characters directly in a macro definition, the command interpreter would not understand them. In macros, these special characters are defined differently:

$L	Redirects the input, similar to <.
$G	Redirects output to a file or a device. A double GG produces the character sequence >>, which directs output to an existing file.
$B	This is the pipe symbol in DOSKEY macros, similar to "|".

To create a macro that sends a directory to the printer, use the following macro definition:

```
DOSKEY PRINTDIR=DIR $G LPT1:
```

By using parameters, you can even create a macro named "Print", which produces a command line to send the output to the printer:

```
DOSKEY PRINT=$1 $2 $3 $4 $5 $G LPT1:
```

In order for the computer to display all of the EXE files in the DOS directory in five columns using DIR, use the command line "DIR C:\DOS\*.EXE /W". To have the computer send the results to the printer instead, use the PRINT macro defined above.

```
PRINT DIR C:\DOS\*.EXE /W
```

When you use DOSKEY remember that it is closely connected to the keyboard and the command interpreter. The macros you created with DOSKEY won't work in a word processor or in a database program. You won't be able to use them in batch files.

Using macro definitions

After you start learning to use macros to speed up and simplify your work, you may become disappointed the next time you start your computer. You'll find that all of your macro definitions are gone and must be re-entered. Obviously, this is time-consuming and defeats the purpose of creating macros - making your work easier.

Although you cannot use DOSKEY macros in batch files, batch files can be used to define DOSKEY macros. For example, to have the PRINT and DIR macros automatically created and available every time you start your computer, type the following command lines in AUTOEXEC.BAT:

```
DOSKEY PRINT=$1 $2 $3 $4 $5 $G LPT1:
DOSKEY DIR=DIR /P $1 $2 $3
```

Alternatively, you can create a special batch file, called MACRO.BAT, which contains all macros you want to use. To activate the macros, call the batch file by entering MACRO from the command line. It doesn't matter whether or not DOSKEY has already been installed.

If you've defined several new macros that you want to keep, use the /MACROS switch so that you don't have to create the MACRO.BAT definition file. Redirect the display of the available macros to a file and then enter the command word DOSKEY prior to each command with an editor.

For example, do this if the MACRO.BAT file hasn't been created and you want to store the existing macro definitions. Start the editor with:

```
EDIT MACRO.BAT
```

For our example macros, the contents of the file would look like the following:

```
DOSKEY PRINT=$1 $2 $3 $4 $5 $G LPT1:
DOSKEY DIR=DIR /P $1 $2 $3
```

Make certain to insert "DOSKEY " at the beginning of each line.

This procedure will write all of the macro definitions into the specified file (MACRO.BAT). The only thing missing is the DOSKEY command. After saving the file and exiting the editor, use these macros by calling the batch file to install them.

DOSKEY editing keys summary

The following is a summary of the DOSKEY key commands:

↑	Provides the previously issued command in the buffer.
↓	Provides the command in the buffer following the displayed command.
←	Moves the cursor one character left.
→	Moves the cursor one character right.
Del	Erases the character to the left of the cursor.
Ins	Shifts back and forth between insert and overstrike mode.
Backspace	Erases the character to the left of the cursor.
PgUp	Makes the first command in the buffer available on the command line.
PgDn	Makes the last entered (most recently used) command available.
Home	Places the cursor at the beginning of the line.
End	Places the cursor at the end of the line.
Esc	Removes the current command line from the screen.
Ctrl+←	Moves the cursor one word to the left.
Ctrl+→	Moves the cursor one word to the right.
Ctrl+Home	Erases all characters (left of the cursor) in the current line to the beginning of the line.

Ctrl+End	Erases all characters (right of the cursor) in the current line to the end of the line.
F7	Displays all of the stored command lines.
Alt + F7	Deletes all of the stored command lines.
F8	Calls a stored line by entering the first letters of a previously used command and pressing F8. DOSKEY searches for the first appropriate line and displays it on the command line for execution or editing.
F9	Specifies a command by the line number in the buffer.
F10	Displays all stored macros.
Alt + F10	Deletes all stored macros.

Special macro characters

$L	Redirects the input, similar to <.
$G	Redirects output to a file or a device. A double GG produces the character sequence >>, which directs output to an existing file.
$B	This is the pipe symbol in DOSKEY macros, similar to "|".
$T	This character is used to separate commands in a macro.
$1-9	Batch parameters, similar to the %1-9 characters in bath files.
$*	This represents all the parameters on the command line.
$$	Represents the dollar sign ($) character.

Quitting DOSKEY

You can't quit DOSKEY like other programs because it permanently remains in the memory of the computer. You can install a new copy with DOSKEY /REINSTALL, but you can't exit any of the copies. To quit DOSKEY and remove it from memory, reboot your computer.

2.6 Using DOS Commands And Files

When you type a command at the prompt, you are asking MS-DOS to perform a simple task for you. You may be asking it to display information or you may be asking it to adjust one or more of its operating settings. In this section, we'll describe some of the characteristics about the DOS commands.

All about commands

In MS-DOS the first entry in a command line is called a command. A space must follow the command. Anything typed after the space is called a parameter(s). This is why commands must be separated on the command line by a space. MS-DOS doesn't differentiate between uppercase and lowercase characters. For example, you can type "FORMAT", "Format" or "format"; for any of these, MS-DOS will execute the FORMAT command.

If you aren't running DOSKEY, leading spaces don't have any affect on the command. So it doesn't matter whether you enter "FORMAT A:" or " FORMAT A:". However, if you created a DOSKEY macro named FORMAT, entering a space at the beginning of the command line would cause the MS-DOS command FORMAT, not the macro FORMAT, to be executed.

Executing commands from any directory

Some MS-DOS commands are in the computer's memory at all times and can be executed immediately. These are called built-in or internal commands. Other MS-DOS commands are first loaded from diskette or the hard drive before they can be executed. These are called external commands. This technique of storing seldom used commands on disk and loading them when needed leaves more room in the computer's conventional memory. Since every command that is stored in memory takes up space, it wouldn't be possible to fit all MS-DOS commands in conventional memory (640K).

When you type an external command at the prompt, MS-DOS searches for it in the current directory of the current drive. If these commands are all contained in the DOS directory, then this procedure will work only if the current directory is the DOS directory. Of course, this isn't always practical especially if you're using a hard drive.

One way to “tell” DOS where, on the hard drive, to find the external command is to specify its path. The path is the drive identifier and the directory where the external command is stored. For example, to use the FORMAT command, which is in the C:\DOS directory, type the following:

```
C:\DOS\FORMAT
```

Having to specify the path for each MS-DOS command quickly becomes cumbersome and annoying. One way to avoid this inconvenience is by using the PATH command.

```
PATH C:\DOS
```

The PATH command tells MS-DOS to search for commands first in the current directory. If it can't find the command there, then it searches for this command in the C:\DOS directory.

So now you can use the FORMAT command from any directory by simply typing the name of the command; the path specification isn't necessary.

You can also tell DOS to search other directories to find commands. Simply use the PATH command and specify these other directories separated by semicolons:

```
PATH C:\DOS;C:\PROGRAM
```

The illustration on the next page shows an example of different directories in a path.

Now MS-DOS searches first in the current directory, then the C:\DOS directory and finally the C:\PROGRAM directory to find a command.

Since the search path specified from the command line is only valid for the current work session, you can put a PATH command in the AUTOEXEC.BAT file. Remember that the search path is always set to the current directory when your computer is first booted. This is why many users put a PATH command in their AUTOEXEC.BAT file.

☞ The PATH command is only used to specify search paths for finding commands.

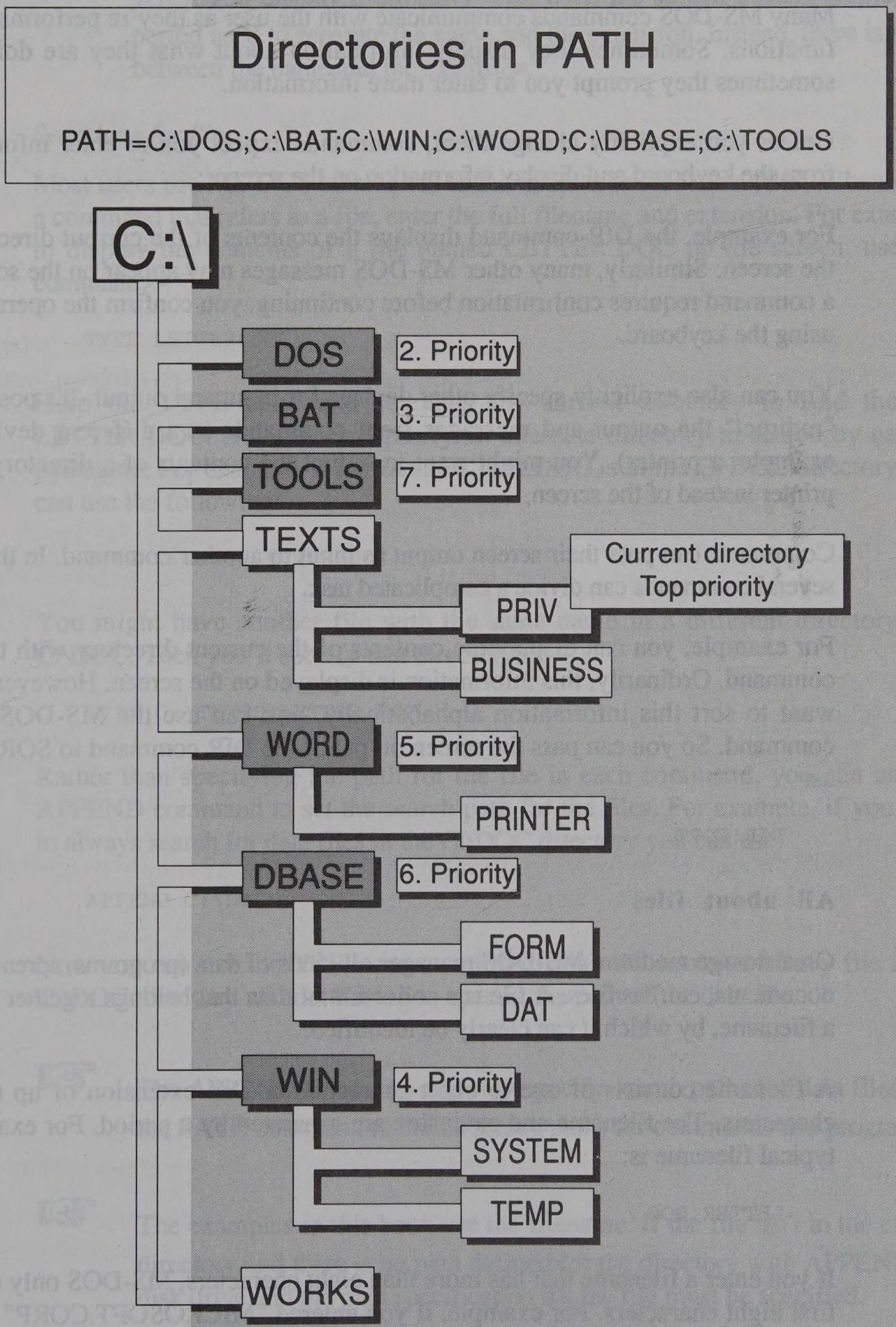
In this book, we use only the command names when executing the commands. If the command isn't in your current directory and you haven't defined a search path for the directory containing the command, you must enter the complete path to the command (the disk drive and directory specifications).

How commands are processed

Most often, you'll enter a command at the keyboard. A part of MS-DOS, called the command interpreter, receives it and first checks whether the command is one of the built-in/internal commands, such as DATE, TIME, etc. If it isn't, the command interpreter first searches in the current directory for an existing program with the .COM extension (especially external DOS commands).

Then the command interpreter searches the current directory for files with the extensions .EXE and .BAT. After this, the search continues with the directories specified by PATH. DOS always searches for commands in this order: COM, EXE and BAT.

If the command interpreter doesn't find an executable file after searching the last directory in the path, the screen displays the message: "Bad command or filename".


Directories in PATH
PATH=C:\DOS;C:\BAT;C:\WIN;C:\WORD;C:\DBASE;C:\TOOLS
C:\
DOS
2. Priority
BAT
3. Priority
TOOLS
7. Priority
TEXTS
Current directory
Top priority
PRIV
BUSINESS
WORD
5. Priority
PRINTER
DBASE
6. Priority
FORM
DAT
WIN
4. Priority
SYSTEM
TEMP
WORKS

An example of using the PATH command

Defining input/output devices for commands

Many MS-DOS commands communicate with the user as they're performing their functions. Sometimes they display information about what they are doing and sometimes they prompt you to enter more information.

Unless you explicitly change them, commands expect you to enter information from the keyboard and display information on the screen.

For example, the DIR command displays the contents of the current directory on the screen. Similarly, many other MS-DOS messages may appear on the screen. If a command requires confirmation before continuing, you confirm the operation by using the keyboard.

You can also explicitly specify other devices for input and output. It's possible to "redirect" the output and messages from commands to a different device (for example, a printer). You might want to output the contents of a directory to the printer instead of the screen.

Commands can pass their screen output as input to another command. In this way, several commands can divide a complicated task.

For example, you can display the contents of the current directory with the DIR command. Ordinarily, this information is displayed on the screen. However, if you want to sort this information alphabetically, you can use the MS-DOS SORT command. So you can pass the screen output of the DIR command to SORT, such as:

```
DIR|SORT
```

All about files

On a storage medium, MS-DOS manages all kinds of data (programs, spreadsheets, documents, etc.) as files. A file is a collection of data that belongs together and has a filename, by which it can clearly be identified.

A filename consists of one to eight characters and an extension of up to three characters. The filename and extension are separated by a period. For example, a typical filename is:

```
LETTER.DOC
```

If you enter a filename that has more than eight characters, MS-DOS only uses the first eight characters. For example, if you entered "MICROSOFT.CORP" as your filename, MS-DOS will shorten it to "MICROSOF.COR".

☞ When you display a directory with DIR, the screen doesn't display the period used to separate the name and the extension. Instead, there is space between the name and the extension.

Searching for files

Most users use MS-DOS commands to manipulate or view data files. When using a command that refers to a file, enter the full filename and extension. For example, to display the contents of a file named LETTER.DOC on the screen, use this command:

```
TYPE LETTER.DOC
```

Here the TYPE command searches the current directory to find the file LETTER.DOC. But you can specify an alternate directory to search by using a pathname. For example, if the file LETTER.DOC is in the C:\DOC directory, you can use the following:

```
TYPE C:\DOC\LETTER.DOC
```

You might have another file with the same name in a different directory, say C:\DOC. Then you'll specify that directory:

```
TYPE C:\TEXTS\LETTER.DOC
```

Rather than specifying the path for the file in each command, you can use the APPEND command to set the search path for the files. For example, if you want to always search for data files in the C:\DOC directory you can use:

```
APPEND C:\DOC
```

Now you can use the TYPE command to display the contents of any file in the C:\DOC directory without having to specify the path of the data file.

☞ The APPEND command can only specify search paths for data files. Use the PATH command to define search paths for commands and programs.

☞ The examples in this book use the filename. If the file isn't in the current directory and there is no path defined for the directory with APPEND, the disk drive and the path specification for the file must be specified.

Be careful when using APPEND. Normally, you should use the complete path of the file instead of APPEND for two reasons:

1. You may get strange results, especially when editing files with application programs. For example, the word processor Microsoft Word 5.0 ignores the search path set with APPEND. Instead, Word 5.0 looks for a specified file in the current directory.

2. You may have problems using APPEND in MS-DOS. While you can display the contents of a file in the file search path easily (TYPE), you cannot delete or copy the file. To delete or copy a file, the directory must be specified explicitly.

Filename naming conventions

As we mentioned earlier, a filename consists of the actual filename, which can have a maximum of eight characters, and an extension of up to three characters. Imbedded spaces aren't allowed. The extension identifies the type of file. Some extensions in MS-DOS are fixed (COM and EXE for executable program files and BAT for batch files), while others have become defacto standards (TXT for text files, DOC for word processor documents, etc.).

All of the letters from A to Z, the numbers 0 to 9 and certain special characters, such as !#$%()&-_, are permitted in filenames. Lowercase letters are automatically converted to uppercase letters.

Characters that aren't allowed in filenames are +=:;.,<>/\ and the space character. The space character is used to separate entries and to separate commands from any additional entries (parameters). Many commands require additional information for correct execution. For example, the FORMAT command won't perform its task of preparing a diskette for storing information unless you at least specify the disk drive as a parameter. The FORMAT command and the disk drive identifier are separated by a space.

Besides these characters, certain groups of characters are reserved for special purposes. Therefore they cannot be used as filenames. Among these are the device names: CON, AUX, COM1, COM2, PRN, LPT1, LPT2, LPT3, NUL.

Specifying more than one file with wildcards

You can execute some MS-DOS commands and use more than one file at a time by using substitute characters (wildcard). These special characters enable you to omit one or more characters from a filename:

*	Substitutes for the rest of the filename or its extension.
?	Substitutes for a single character in the filename or its extension.

Wildcards, which simplify working with several files, are required to execute some tasks.

Wildcards have two very important jobs in MS-DOS:

- They can be used to indicate several files. For example, copy all of the files, with the .EXE extension, from the C:\DOS directory without using a wildcard. In order to do this, you would have to determine all of the filenames and then enter a command line with the MS-DOS COPY command for each file. It's a lot easier if you use a wildcard to copy all of the files with the .EXE extension: *.EXE.

- Wildcards are also useful if you want to filter information. Suppose that you're looking on the hard drive for a file named LETTER.TXT. You could use DIR LETTER.TXT to search for the file; by entering the /S switch you would include the subdirectories in your search as well. If you didn't know the exact name of the file you were looking for, you obviously wouldn't want to go through LETTER.TXT, LETTER1.TXT, LETTERS.TXT etc. It's much easier to type the part of the filename that you do know and use a wildcard for the rest: LETTER*.TXT.

The "*" character in filenames or extensions indicates that the remainder of the filename or its extension can be any combination of characters. Any character you type after the "*" is ignored.

Examples

A*.EXE	All filenames beginning with A and ending with the .EXE extension (APPEND.EXE, ATTRIB.EXE).
M??ER.TXT	All files with 5 letters in the filename and the .TXT extension. The first letter is M and the last two letters are ER (MEIER.TXT MAYER.TXT, MEYER.TXT).
LETTER.*	Files named LETTER that have any extension.
INVOICE.*	All files named INVOICE that have any extension (INVOICE.TXT, INVOICE.DOC, INVOICE.SIK etc.).
.	All files.

Use wildcard characters carefully when referring to more than one file. The command could affect files that you really didn't want to include, especially when you're deleting files. Therefore, first use DIR to display the files that should be included. For example, if you want to delete all of the files in a directory that begin with the letter A, simply type:

```
DIR A*.*
```

This displays the names of all of the files that meet your selection criteria. When you're certain that the file specification is correct and that you no longer need any of the files, type:

```
DEL A*.*
```

Naturally you'll want to be careful when you type a command such as the one above. You may end up deleting files that you didn't intend to delete.

Unfortunately, not all MS-DOS commands work with wildcard characters. For example, the TYPE command will only display the contents of a specific file on the screen. Entering TYPE *.TXT won't display the contents of all of the text files.

In many cases you can write a batch file to accomplish the same tasks as using wildcards. You could also use another command with wildcards that will have the same effect.

For example, we already mentioned that TYPE *.BAT will not display all of the BAT files in the current directory. But you can use a variation of the COPY command to accomplish the same thing. Here you can copy all of the BAT files to the screen (called CON: for the console device) using this:

```
COPY *.BAT CON
```

2.7 Working With Disk Drives And Hard Drives

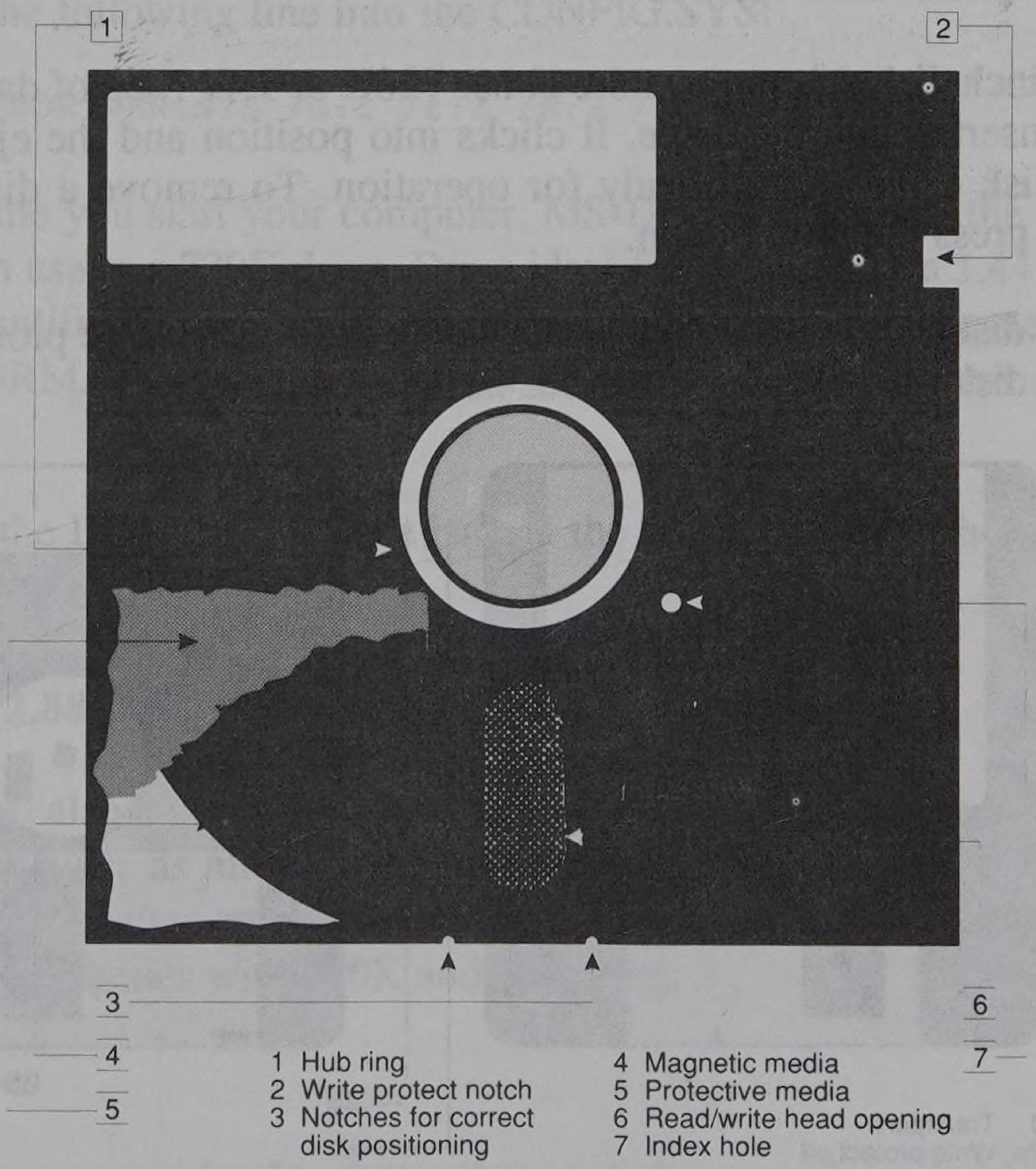
This section discusses some things you should remember when working with disk drives and hard drives.

5 1/4 inch disk drives

There are two kinds of 5 1/4 inch disk drives. One model can store up to 360K of data. A high capacity model can store up to 1.2 Meg of data.

To use a diskette, insert it in the disk drive and close the drive door. The disk drive is then ready for operation.

To remove a diskette from a disk drive, lift the door lever and pull out the diskette. You should immediately place the diskette in its paper jacket (sleeve) to prevent it from getting dirty or being touched through the opening, which exposes the media itself. To protect diskettes from being accidently deleted or modified, cover the notch on the side of the diskette with tape. This is usually referred to as write protecting the disk.


1
2
3
4
5
6
7
1 Hub ring
2 Write protect notch
3 Notches for correct disk positioning
4 Magnetic media
5 Protective media
6 Read/write head opening
7 Index hole

The main parts of a 5 1/4 inch diskette

You should protect diskettes from high temperatures as well as electromagnetic fields. Leaving a diskette in your car on a sunny day could destroy it and your data.

If you leave a diskette near magnets or transformers, such as those found in computer monitors, your data may also be destroyed.

☞ You can write to both sides of a diskette without having to remove it from the disk drive and turning it over. The disk drive is capable of writing on both sides of the diskette.

360K disk drives

DSDD diskettes (Double Sided, Double Density) are used to store up to 360K of data.

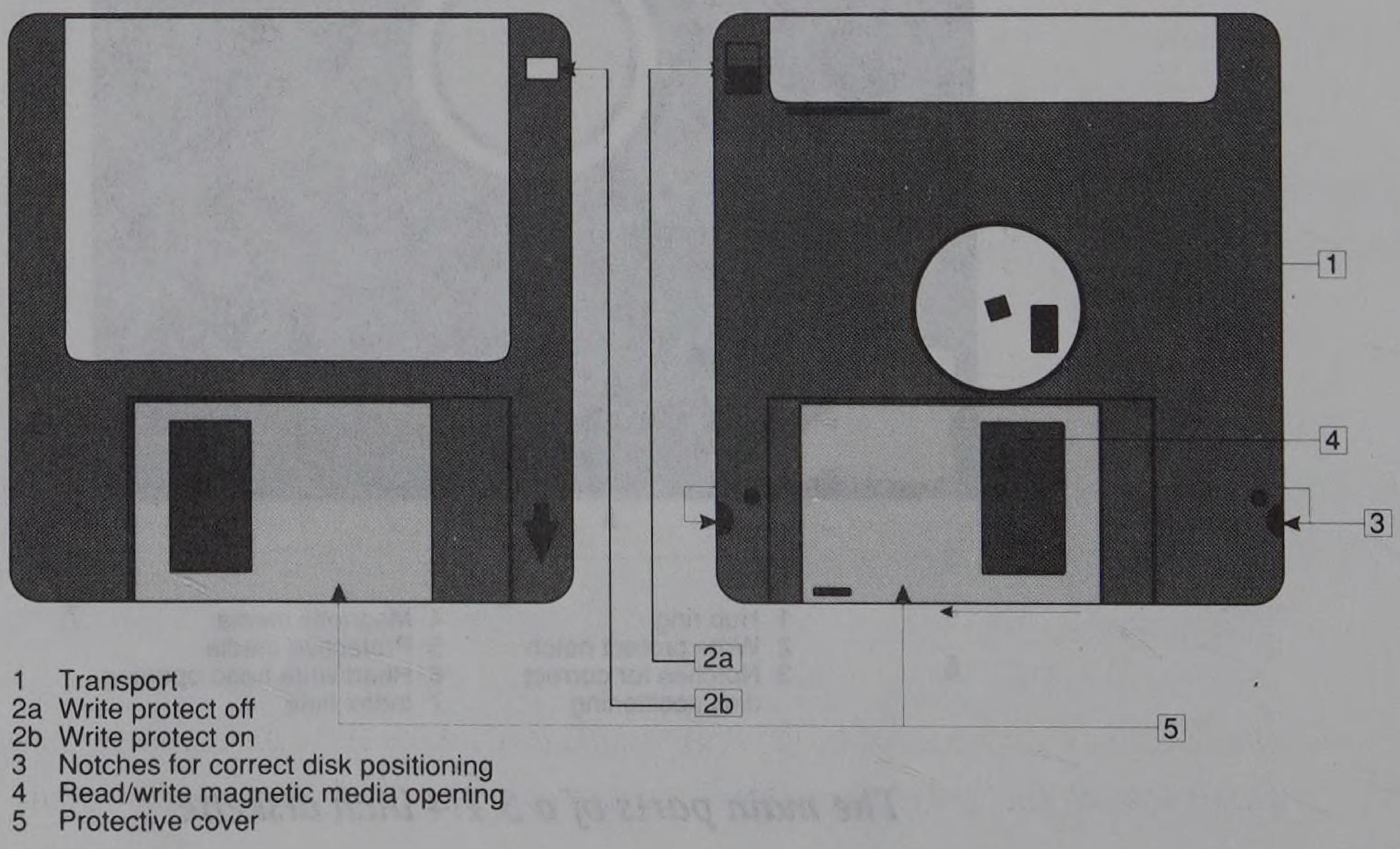
1.2 Meg disk drives

DSHD diskettes (Double Sided, High Density) are used to store up to 1.2 Meg of data. You can also read, write and format 360K diskettes with a 1.2 Meg disk drive. The disk drive automatically recognizes when a 360K diskette is inserted. To format a 360K diskette in a 1.2 Meg disk drive, use a special switch (/4) as a parameter for the MS-DOS FORMAT command.

3 1/2 inch disk drives

3 1/2 inch disk drives can store either 720K or 1.44 Meg of data. To use a diskette, first insert it into the drive. It clicks into position and the eject button pops out. The disk drive is then ready for operation. To remove a diskette from the disk drive, press the eject button.

These diskettes can be write protected by sliding the write protect notch on the side of the diskette so that you can see thru the opening.



1 Transport
2a Write protect off
2b Write protect on
3 Notches for correct disk positioning
4 Read/write magnetic media opening
5 Protective cover

The main parts of a 3 1/2 inch diskette

☞ To protect the data on a 3 1/2 inch diskette from being deleted or overwritten, slide the plastic write protect notch so that you can see through the hole.

720K disk drives

DSDD diskettes (**D**ouble **S**ided, **D**ouble **D**ensity), are used to store up to 720K of data.

1.44 Meg disk drives

DSHD (**D**ouble **S**ided, **H**igh **D**ensity) diskettes are used to store up to 1.44 Meg of data. You can also read, write and format 720K diskettes with a 1.44 Meg drive. If you're going to use the disk drive only for diskettes with 720K, use the DEVICE=DRIVER.SYS command in the CONFIG.SYS with the optional parameters or the SETUP program for your computer (NOT the MS-DOS SETUP program).

For example, to use the second disk drive B: with 720K instead of 1.44 Meg, you must type the following line into the CONFIG.SYS:

```
DEVICE=DRIVER.SYS /d:1 /f:2 /H:2 /S:9 /T:80
```

The next time you start your computer, MS-DOS will display the drive identifier that you can use as a 720K drive. Drive identifier B: remains a 1.44 Meg drive, but the new identifier acts like a 720K drive. You won't have to enter any switches with the FORMAT command to format 720K diskettes on the new drive.

☞ If the DRIVER.SYS file isn't in the main directory, the entire path must be specified.

The new 2.88 Meg disk drives

MS-DOS 5.0 also supports the new 3 1/2 inch disk drives with a capacity of 2.88 Meg. This is twice as much as the previous capacity. The new disk drives needed for 2.88 Meg diskettes are downwardly compatible, which means that they can also be used with diskettes with 720K and 1.44 Meg.

Hard drives

Hard drives are considerably faster than disk drives and can hold a lot more data. The capacity depends on the hard drive; the minimum is generally 20 Meg or 30 Meg.

Differences from diskettes

Hard drives differ from diskettes in the following ways:

- While a diskette spins in the disk drive at a rate of 300 revolutions per minute, the storage medium in a hard drive spins at 3600 revolutions per minute. The higher speed (twelve times higher) makes it possible for the hard drive to read and write data much faster.
- The number of tracks and sectors per storage medium is increased significantly. Also, the distance between the read/write head and the storage medium is much smaller. This accounts for the higher storage capacity.
- A hard drive doesn't consist of one disk spinning. Hard drives are made up of stacks of 2 to 6 disks, with two read/write heads for each disk. All disks in the stack spin simultaneously. This also accounts for the higher storage capacity.

Hard drives are sensitive to shock. Sometimes computer systems are delivered with a PARK command. PARK moves the heads of the hard drive to a special "Safe" position. Before you move your computer you should run the PARK program. Since MS-DOS itself does not have this program, you should use the program that is supplied by the manufacturer of your hard drive.

Using a hard drive: Special features

Most commands that can be used with a diskette also work on a hard drive. The commands DISKCOPY and DISKCOMP for copying and comparing diskettes are exceptions. You cannot use these commands on a hard drive.

Unfortunately a hard drive cannot be write protected like a diskette. In a sense, the data on a hard drive is "unprotected". However, instead of write protect, certain precautions can be taken to protect the data on a hard drive. Files can be given a special attribute that protects the file from being accidentally destroyed or changed (write protect attribute).

2.8 Starting Application Programs

Application programs can be started at the command line, just like other MS-DOS commands.

Loading application programs from a diskette

To run a program from a diskette, insert the diskette that contains the application program in drive A: and make this disk drive the current drive by typing:

```
A: Enter
```

Now enter the name of the application program and press Enter.

☞ If you don't know the name of the program, use the DIR command to display all of the files on the diskette. Executable files have .EXE, .COM or .BAT extensions. For example, you can use "DIR *.COM" to display all of the COM files. Program names are often abbreviated with two letters. For example, the filename for Multiplan is abbreviated as "MP".

Starting applications from the hard drive

If you define a search path for the directory of the application with the PATH command, you can start the application by entering the name of the program.

If you didn't define a path, you can make the hard drive the current drive by typing:

```
C: Enter
```

Then use CD and the name of the directory to change to the directory of the application program:

```
CD Directory_name Enter
```

Start the application by entering the name of the application program. For example, if the word processor Word 5.0 is in the WORD directory on the hard drive, select the directory by entering CD \WORD. If you're currently in the main directory, you won't need to enter the backslash; instead enter CD WORD.

With MS-DOS commands you can specify the entire path of the program. So, you don't have to change directories. For example, the Word word processor, which is stored in the C:\WORD directory, can be started from any directory by typing the following command line:

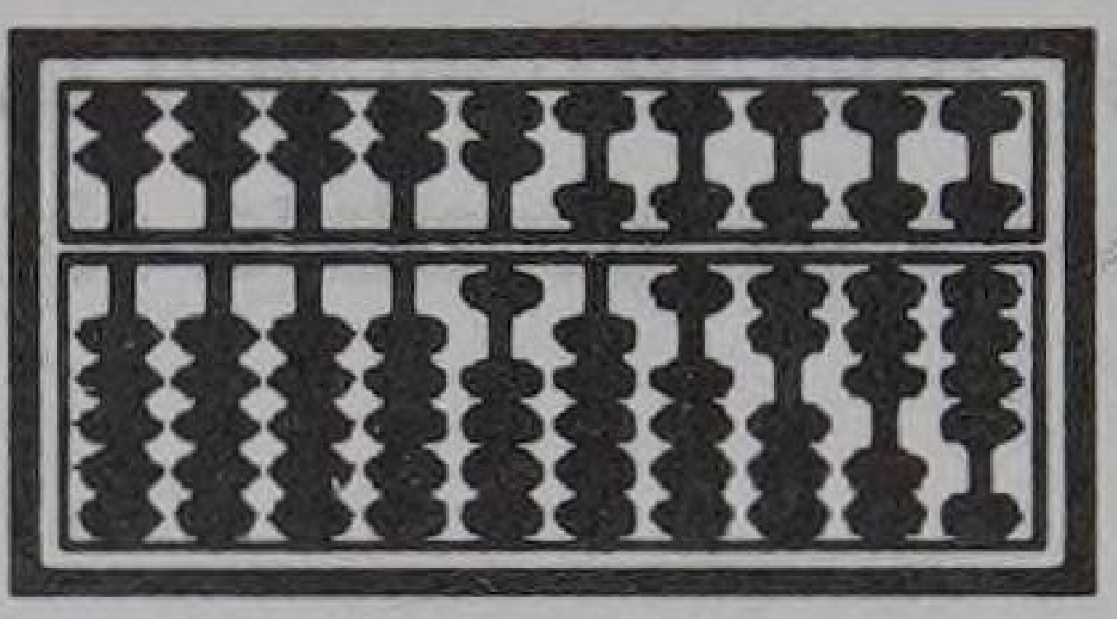
```
C:\WORD\WORD
```

☞ Some programs expect that you'll run them from their own directory (the program directory must be the current directory). You can solve this problem easily with a batch file, in which one of the commands changes to the appropriate directory and the next command starts the program. (Refer to the hints in this book.) For example, to start Word from the WORD directory, create the WORD.BAT batch file:

```
C:
CD C:\WORD
WORD %1 %2 %3
```

If you place this batch file in a directory which is in the search path (for example, in the directory of other DOS commands), then you can run the word processor.

Chapter 3



Storage Devices

Computers are used to create, change, edit and manipulate data in various forms. Data is manipulated in the computer's memory. But this memory is volatile - the contents, and therefore the data, is lost when you switch off the computer. So a permanent way of storing the data is required.

3.1 Working With Storage Media

Let's start out with a warning - use any storage media with care. Any damage to a diskette, such as high temperatures or magnetic fields, may cause loss of data. Hard drives are better protected than diskettes because the actual storage medium is hermetically sealed. However, even hard drives are still sensitive to shock.

Remember the following when working with floppy diskettes:

- Always keep 5 1/4 inch diskettes in their envelopes (sleeves).
- Never touch the magnetic surface of diskettes.
- Do not bend diskettes; apply as little pressure to them as possible.
- Use only felt pens to write on diskettes. It's better to first write on the label, then apply the label to the diskette.
- Do not expose diskettes to temperatures over 120 degrees Fahrenheit (about 50 degrees Celsius). Leaving a diskette in your car during a hot summer day may damage the media.

- Keep diskettes and hard drives away from electromagnetic fields. Don't place diskettes on or next to the monitor or near transformers or power supplies.

- Protect hard drives from vibrations; ensure that the computer is safely positioned on a sturdy desk or table.

- Never switch off the computer when your hard drive or diskette is in operation; disk drives and most hard drives have indicator lights that are lit during read/write accesses.

3.2 Different Storage Devices

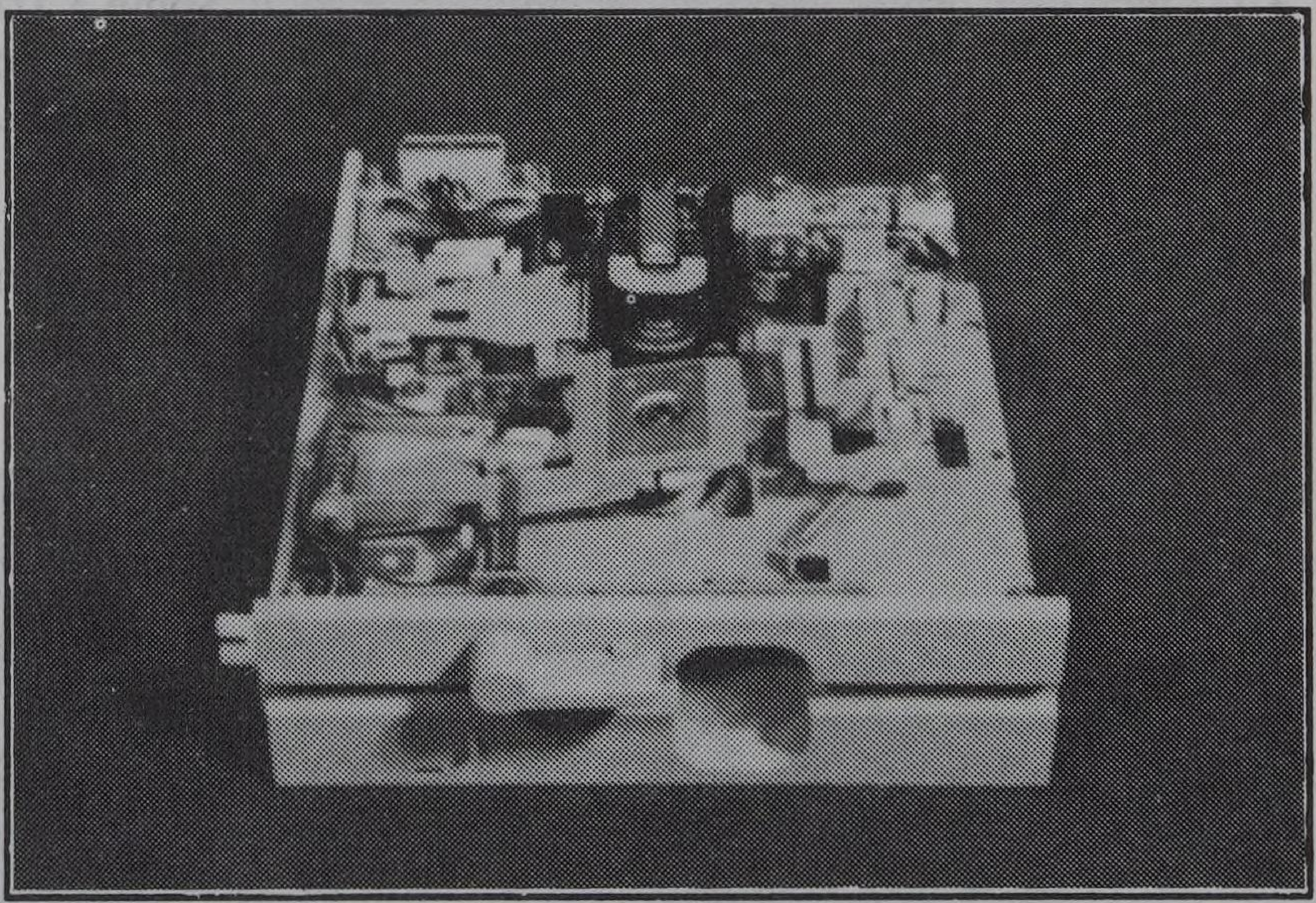
The quality of diskettes has been continually improving. There are several kinds of diskettes and methods of storage available. There are two physical sizes of disk drives, each with different capacities.

5 1/4 inch disk drives

This disk drive is for the diskettes made of flexible plastic with a dimension of 5 1/4 inches.

A standard PC/XT computer has a 360K disk drive, capable of holding 360,000 characters of data or about 150 typewritten pages. The diskettes are labelled DSDD (**D**ouble **S**ided/**D**ouble **D**ensity).

A standard AT computer has a 1.2 Meg disk drive, capable of storing 1,200,000 characters of data or about 500 typewritten pages. This higher storage capacity is possible because of a higher recording density and special diskettes. These higher density diskettes are identified as DSHD (HD = **H**igh **D**ensity). You can't determine whether a diskette can store 360K or 1.2 Meg by simply looking at it.



A 5 1/4 inch disk drive

Although a high density diskette can be inserted into a 360K disk drive, you won't be able to read or write to the diskette. However, an AT can read both 360K and 1.2 Meg diskettes. To format a 360K diskette in a 1.2 Meg disk drive, you'll need to use a special parameter on the FORMAT command. In this case, the disk drive will only write on every other data track, which produces the 360K capacity.

If you are using an AT, you should have a supply of each of these diskettes. Since most programs are stored on 360K diskettes, you can use the less expensive 360K diskettes to create backup copies.

There are two disadvantages to using 5 1/4 inch disk drives. They are rather large and take up a lot of space in the computer cabinet (which is always getting smaller). Also, the floppy diskettes are relatively unprotected. Since the inner magnetic "donut" (which stores the data) is visible in several places, it can be easily damaged. These floppy diskettes can also be accidentally bent. 3 1/2 inch diskettes don't have these disadvantages.

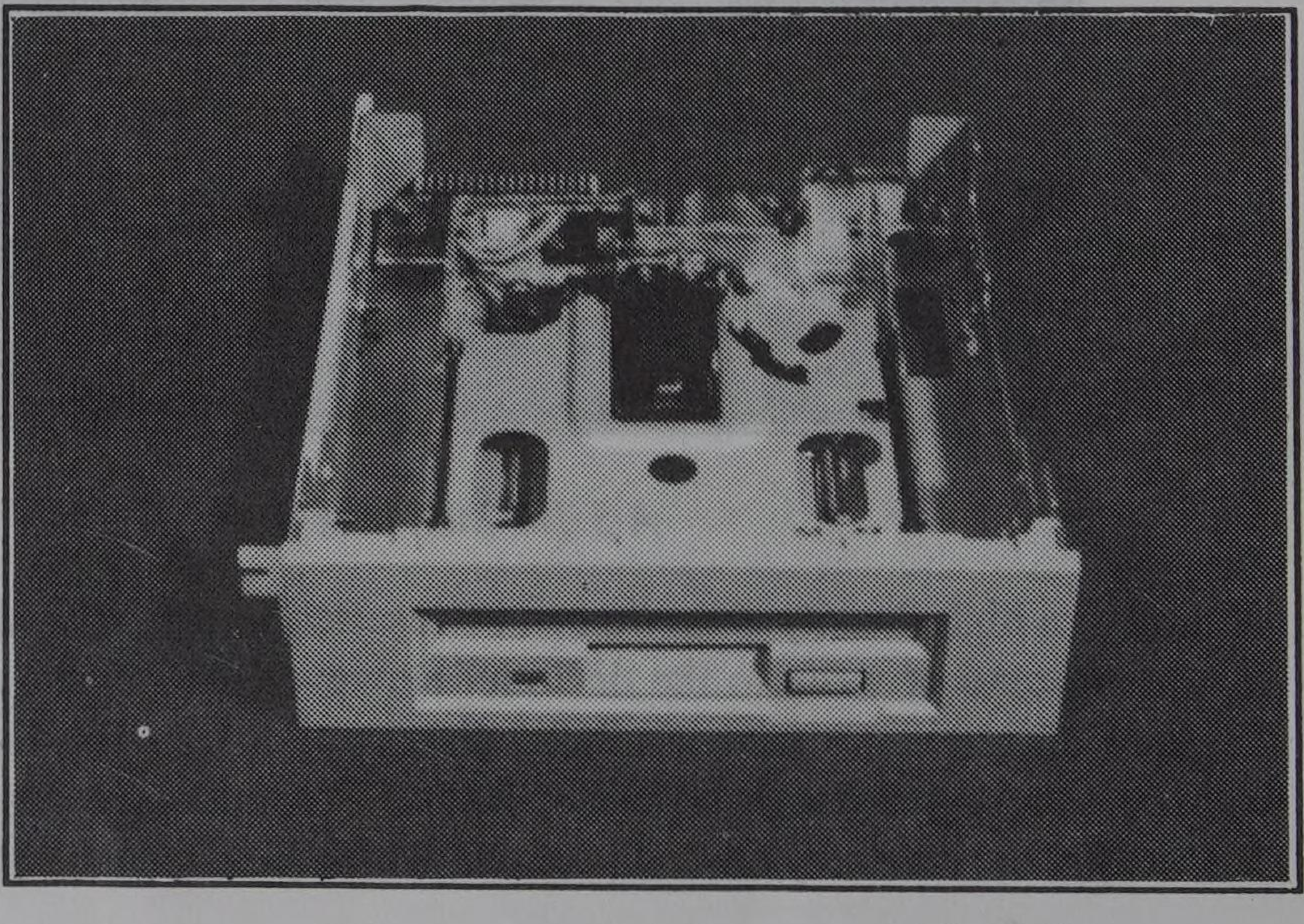
3 1/2 inch disk drives

3 1/2 inch disk drives are standard in most portable and laptop computers and on many desktops as well.

These diskettes are made of hard plastic, which have a width of 3 1/2 inches. The inner magnetic donut is hidden by a metal slide that remains closed until the diskette is placed in the disk drive. Removing the diskette from the drive closes the slide. This type of diskette is well-protected from damage and dirt. For example, you can safely carry 3 1/2 inch diskettes in your shirt pocket.

3 1/2 inch disk drives also vary in storage capacity. Some 3 1/2 inch diskettes can store 720K (double density on DSDD diskettes) while others can store 1.44 Meg (on special DSHD diskettes). The high density DSHD diskettes are more expensive but the increased storage capacity is usually worth the expense.

A 1.44 Meg disk drive can also read and use diskettes with a lower capacity. So if you buy a 1.44 Meg disk drive for a little more money, you can still use 720K diskettes.



A 3 1/2 inch disk drive

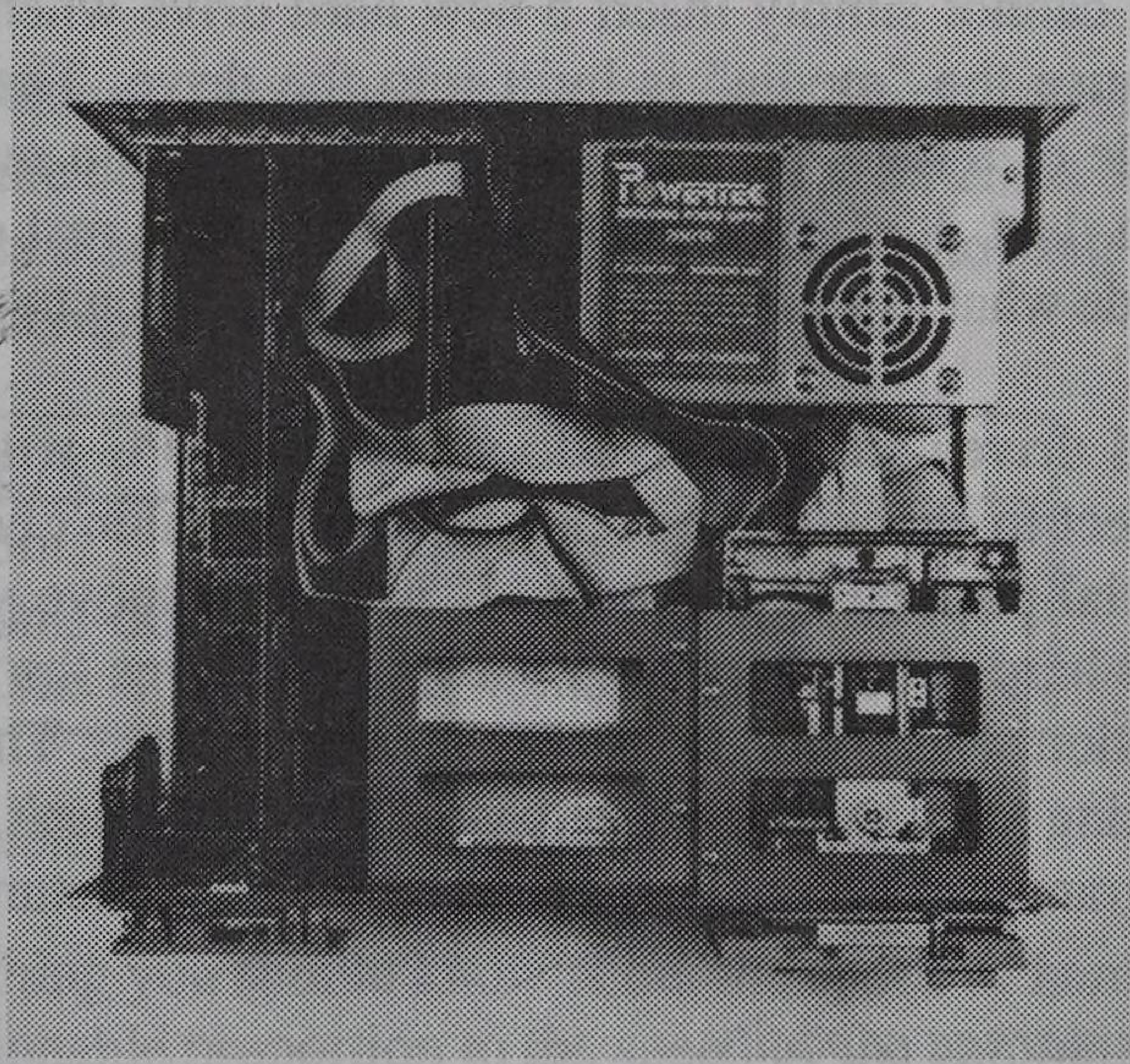
The new 2.88 Meg diskettes

A new higher capacity 3 1/2 inch disk drive is now available. These new drives are capable of storing up to 2.88 Meg of data, which is twice as much as the previous storage capacity. Since the new disk drives are downwardly compatible, they can also be used to read and write data to/from 720K diskettes and 1.44 Meg diskettes.

Right now these new diskettes and disk drives are quite expensive. But when more units are available on the market, their price will drop. In the mean time, MS-DOS 5.0 is prepared to accommodate the new disk drives and diskettes.

About hard drives

In the beginning, PC's came with two disk drives. Today, most are equipped with a hard drive.



A PC containing a hard drive

From the outside a hard drive just looks like a large black box. Besides numerous electronic integrated circuits, there isn't very much to see. It's necessary to package the hard drive like this in order to achieve such high data storage density. The mechanical tolerances have to be absolutely high and the mechanism has to be completely dust and dirt free.

The storage media consists of several platters (like phonograph records) arranged in stacks and sealed in a metal housing. They are called hard drives because the platters are made of a rigid metal instead of pliable mylar.

Hard drives offer users certain advantages over disk drives.

Advantages of hard drives

- A lot of data can be stored on a hard drive. The normal 20 Meg hard drives can store about 9,000 typewritten pages, which is the contents of more than 60 individual 360K diskettes.

- Reading data from and writing data to a hard drive is much faster than to a disk drive. This speeds up your work, especially with programs that access the disk drive often or require large amounts of data.

However, there are also certain disadvantages in using a hard drive.

Disadvantages of hard drives

- You cannot simply insert a new hard drive when your old hard drive is full because the "media" is built into the drive. The reason for this is that the inside of a hard drive and the storage medium must be completely protected from dust. It's hermetically sealed from the rest of the world. So users must decide what to store on the hard drive and when to delete unneeded files.

- Hard drives are sensitive to shock. So a PC with a hard drive shouldn't be subject to shock during operation and shouldn't be transported without taking special protective measures. If you must move your PC, you should "park" your hard drive. Parking a hard drive moves the read/write head to a special position where vibrations won't cause damage. There are special programs for parking your hard drive, but MS-DOS 5.0 doesn't have such a program. These programs are generally supplied by the manufacturer of the hard drive.

Selecting storage capacity

When you buy a hard drive, there are several items to consider. First is the storage capacity of the hard drive. The computer systems on the market today usually include a 20 Meg hard drive for about $300 to $400. This is usually sufficient for personal use. If you save only the data, which you use frequently, on the hard drive, you can store rarely used data and programs on diskettes.

If you're going to use your PC for business or anticipate a large amount of data, a 20 Meg hard drive might not be sufficient. Although hard drives are available with up to several hundred Meg of capacity, these drives are quite expensive. If you just need a little more storage space, the most reasonable alternative is to install a 40 Meg drive, which adds a few more dollars to the purchase price of your computer.

You'll soon see how quickly the hard drive can fill up with data. The more data stored on your hard drive, the slower it works. So you should never fill your hard drive to more than 90 percent of its total capacity. This would drastically reduce the speed of your computer because of the way data is stored.

The speed of a hard drive

The sticker on your hard drive usually lists both the capacity and access speed. For example, you might see the following:

```
Hard drive 20 MByte 65ms
```

The "ms" represents milliseconds, which is 1/1000 seconds. This refers to the "access time", which is the average amount of time the hard drive needs to read information stored on it. The higher the number, the longer it takes the hard drive to read and the slower the hard drive works.

Hard drives with 40ms - 60ms are about average. Hard drives with greater than 60ms access time are considered slow. Hard drives with 28ms to 40ms are fast; an access speed of less than 28ms is extremely fast. The following are some common values for various computer models:

Access	PC/XT	AT	386
Maximum	65ms	40ms-60ms	40ms
Typical	40ms-60ms	40ms	28ms-40ms
Optimal	40ms	28ms-40ms	<28ms

Table of hard drive access times

There's one more thing you should know about the speed of hard drives. The speed should be in a reasonable proportion to the speed of the computer clock frequency. For example, a hard drive rated under 40ms is unnecessary on a very slow PC/XT with a clock rate of 4.77 Mhz. However, a slow hard drive rated at say 60ms will slow down a fast 386.

In order to have an efficient computer system, you must have the right combination of individual components. Although the speed of a hard drive depends mostly on mechanical and electrical components, MS-DOS can increase the performance within certain limitations. If you have enough memory available, you can change the parameters of the BUFFERS, FASTOPEN and SMARTDRV commands to increase the performance of your hard drive.

☞ For a reasonable price you can easily install a second hard drive in most PCs. Most computers already have a connection for a second drive.

Although access is faster to a RAM disk, which is really just a reserved portion of the computer's memory, the contents of a RAM disk are lost when you switch off your computer. So anything saved in a RAM disk must also be copied to a hard drive or a floppy diskette before switching off the computer.

The second fastest is the hard drive. In this case, a typical 40 Meg drive is divided into two partitions (one of 32 Meg and a second partition of 8 Meg) in order to work with MS-DOS 3.3. MS-DOS 5.0 can manage much larger capacity hard drives without dividing them into partitions.

3.3 Formatting Diskettes

Data can be written only to storage media that has been previously "prepared" or formatted. This applies to hard drives as well as the various kinds of floppy diskettes. One exception is the RAM disk, in which can store data after you start its driver program from the CONFIG.SYS file.

When you purchase a diskette, you receive a storage medium that is coated with magnetic material. Before you can use the diskette, it must be formatted. Formatting a diskette divides it into tracks which are further divided into sectors. MS-DOS uses these sectors to store "chunks" of data. MS-DOS also creates special reserved areas on the media for managing the stored information, similar to a root directory. The MS-DOS FORMAT command is used to format diskettes.

Preparing a diskette: formatting

Dividing the diskette into tracks and sectors and preparing information for managing the diskette take place during formatting. All media must be formatted before it can be used.

During FORMAT, MS-DOS checks the diskette for defective areas and marks these bad areas so that they aren't used later to store data.

Since bad sectors are marked, a diskette may not be able to hold the normal maximum capacity after it's formatted. Information about bad sectors is displayed when the formatting procedure is completed. The CHKDSK command can also be used to verify how much storage space is available on a diskette.

When you format a diskette, it's always formatted to the greatest possible capacity of the disk drive, unless you specify otherwise. So a 1.2 Meg disk drive is always formatted at 1.2 Meg capacity.

☞ Since FORMAT deletes all data from the diskette, you can also use this command to reformat diskettes that already contain data. However, before formatting such a diskette, use the DIR command to view the contents of the diskette. Then you can decide whether you really want to delete the data on it.

Special features of formatting

Formatting a diskette is a mechanical and magnetic process. So you can only use the FORMAT command on disk drives that have the corresponding mechanism and electronic circuitry. This means that you cannot specify a pseudo disk drive defined by ASSIGN, SUBST and JOIN or a drive in another computer on a network. Likewise, you cannot use FORMAT to prepare a RAM disk (a pseudo disk drive in main memory). Instead a RAM disk is "installed" when you define it to MS-DOS.

☞ FORMAT prompts you to insert the diskette to be formatted and to confirm your action by pressing Enter. You have the option of starting the command from the DOS diskette and then inserting the diskette to be formatted into the same disk drive.

Formatting with different capacity disk drives

When formatting diskettes, you must distinguish between different capacity disk drives.

360K disk drive (5 1/4 inch)

A 360K disk drive processes double sided, double density diskettes (DSDD). When you use the MS-DOS FORMAT command to prepare these diskettes on a 360K disk drive, you don't have to specify any additional parameters.

```
FORMAT driveid
```

For example, to prepare a diskette in drive A: (first disk drive) to store data, use the following command line:

```
FORMAT A:Enter
```

The FORMAT command prompts you to insert the diskette to be formatted. After you press Enter, formatting begins, displaying the progress on the screen. Finally, MS-DOS 5.0 prompts you to enter a volume label for the diskette. You can enter a label of up to eleven characters or simply press Enter to skip this option:

☞ You cannot use high density diskettes (1.2 Meg diskettes) in a 360K disk drive.

1.2 Meg disk drive (5 1/4 inch)

A 1.2 Meg disk drive is a high capacity version of a 360K disk drive. These drives can store more than three times as much as a 360K diskette, but require special DS HD (double sided, high density) diskettes.

To format a high density diskette, use FORMAT without adding any parameters:

```
FORMAT driveid
```

For example, to format a high density diskette in drive A:, use the following command line:

```
FORMAT A:Enter
```

Normally, when a disk drive is used, it automatically recognizes what capacity diskette has been inserted. However, this doesn't happen when you format a diskette. By default, a 1.2 Meg disk drive formats a diskette at the highest possible capacity (1.2 Meg). To override this default, enter the /F switch, a colon and the desired capacity after the command and the disk drive specification. In our example, this would be :360.

To format a 360K diskette in a 1.2 Meg disk drive, use the following command line:

```
FORMAT A: /F:360
```

The /F switch was new to the FORMAT command beginning with MS-DOS 4.0. In previous versions of MS-DOS, you specified a 360K capacity as a /4 switch. To remember this switch, remember that a 360K diskette has only about 1/4 of the capacity of a 1.2 Meg diskette. So an alternative way to format a diskette with a 360K capacity is:

```
FORMAT A: /4
```

If you don't enter the switch (/4 or /F:360), the disk drive will try to format the diskette at a higher capacity than is allowable. But trying to format a DSDD diskette at a high capacity will almost always result in errors because the media is not designed for this density. As a result, formatting takes longer because of the errors.

One way to "bypass" the /F and /4 switches is to use a feature of MS-DOS 5.0 to set up multiple drive identifiers for the same disk drive.

To do this, insert a special DRIVER.SYS statement into the start up CONFIG.SYS file. Here's an example that sets the A: drive, normally a 1.2 Meg, to have a default of 360K when identified as the D: drive:

```
DEVICE=C:\DOS\DRIVER.SYS /D:0 /F:0
```

The instructions have the following meanings:

DEVICE=C:\DOS\DRIVER.SYS

> This starts the driver program. You may have to specify a different directory path instead of C:\DOS.

/D:0	This specifies the actual disk drive to be used. 0=Disk drive A:, 1=Disk drive B:.
/F:0	This determines the kind of disk drive, 0=360K, 1=1.2 Meg, 2=720K, 3=1.44 Meg.

The next time you boot the computer, the driver program will display a message similar to the following (instead of D:, it may specify another disk drive identifier):

```
Loaded external Disk Driver for Drive D:
```

Now you can address drive A: as a 1.2 Meg disk drive and address the same disk drive under the label D: as a 360K disk drive. On drive D: you won't need to add "/F:360" when you format a 360K diskette. It's the same physical disk drive, but it can be addressed by two different drive identifiers, A: and D:.

Problems exchanging data between ATs and PCs

One reason 1.2 Meg disk drives are popular is that they can also access 360K diskettes. However, few people realize that compatibility can only be guaranteed when reading 360K diskettes. There is no guarantee that you can write to them error free.

This happens because a 1.2 Meg disk drive can write 80 tracks. So its tracks are narrower. But when an XT type drive writes its "wide" tracks on the same diskette, there is magnetization at the edge of the tracks that could cause read errors.

720K disk drive (3 1/2 inch)

Three and one-half inch disk drives are standard for all laptop computers and for many of the newer desktop computers. These drives are more sturdy than 5 1/4 inch disk drives. Even though they have advantages, most computers still use 5 1/4 inch disk drives.

The smaller 3 1/2 inch diskettes are constructed differently from its 5 1/4 inch counterpart. The diskette case is made of sturdy plastic and instead of a hole in the middle, it has a small circular metal piece that spins the media. The read/write slot is protected by a metal slide. The slide opens automatically when you insert the diskette into the drive and closes when you remove the diskette from the drive.

The 3 1/2 inch diskettes are a lot sturdier and less susceptible to physical damage than 5 1/4 inch diskettes. You must protect 5 1/4 inch diskettes from dust, humidity and touch. Although 3 1/2 disks can also be damaged by water or dust, they are better protected due to their sturdy cases. As you can see, 3 1/2 inch diskettes have several advantages.

When you buy 720K 3 1/2 inch diskettes, they should be double sided (DS) double density (DD).

You can format 720K diskettes without specifying any additional parameters:

```
FORMAT driveid
```

If you use a 3 1/2 inch disk drive with 720K capacity as your second disk drive (B:), use the following command to prepare your diskette:

```
FORMAT B:
```

1.44 Meg disk drive (3 1/2 inch)

Once again it was IBM who brought a new development on the market. The new PS/2 models are equipped with a 3 1/2 inch disk drive with 1.44 Meg capacity.

They can be used at the 1.44 Meg high capacity or the 720K standard capacity.

To prepare a 3 1/2 inch high density diskette for data (1.44 Meg), type the following command for disk drive B:

```
FORMAT B:
```

Since high density (HD) 1.44 Meg diskettes are fairly expensive (about $20 for 10), you may choose to use the less expensive, standard DSDD diskettes which can be used to store up to 720K of data. Then you can format them using one of the following ways:

Option 1:

With MS-DOS 5.0 the procedure for formatting a double density disk is easy. Simply enter /F:720 after the drive identifier when you format the diskette. This tells your 1.44 Meg disk drive to format the diskette at 720K capacity.

```
FORMAT B: /F:720
```

Option 2:

Insert a line in the CONFIG.SYS file so that your disk drive is also available under a new disk drive identifier with 720K. The following is the command to install a 1.44 Meg disk drive installed as disk drive A:

```
DEVICE=C:\DOS\DRIVER.SYS /D:0 /F:2
```

If the second disk drive (B:) on your computer is the 3 1/2 inch drive, replace "D:0" with "D:1". The next time you start your computer the DRIVER.SYS program will display a message telling you which drive identifier (letter) you can use as a 720K disk drive. Remember that the original disk drive identifier (letter) for use with 1.44 Meg diskettes is still valid.

For example, if you installed a hard drive on your computer, it will have drive letter C:. The DRIVER.SYS program then assigns the newly defined disk drive the letter D:.

If you installed the 3 1/2 inch disk drive as a second disk drive (B:), you can still format 1.44 Meg diskettes with the following:

```
FORMAT B:
```

When you want to create a 720K diskette, simply use your new drive identifier (drive D:):

```
FORMAT D:
```

1.44 Meg diskettes must also meet certain quality requirements: DSHD (double sided, high density) appears on the label. Although the 1.44 Meg disk drive recognizes lower density diskettes (DD) you won't be able to format them at 1.44 Meg.

It's only a matter of time before 3 1/2 inch diskettes with 1.44 Meg memory capacity becomes popular. After all, they're not much more expensive than 720K diskettes. If you buy an AT and want to use a 3 1/2 inch disk drive, we recommend that you also buy a 1.44 Meg disk drive.

2.88 Meg disk drive (3 1/2 inch)

MS-DOS 5.0 is prepared for the new 3 1/2 inch diskettes and disk drives with 2.88 Meg capacity. The capacity is doubled compared to the 1.44 Meg disk drives. This is done by doubling the sectors per track from 18 to 36.

Since MS-DOS automatically recognizes the capacity, special preparations aren't necessary in order to use diskettes with lower capacities in this type of disk drive.

To format 720K diskettes, use /F:720 as a switch. To format 1.44 Meg diskettes use either one of these switches: /F:1.44 or /F:1440.

Formatting diskettes as system diskettes

When you're formatting a new diskette and want to copy the operating system onto the diskette, use the /S switch. This switch copies two hidden files and the command interpreter COMMAND.COM onto your diskette. This leaves a little less storage space on the diskette. If you don't have MS-DOS installed correctly on your hard drive, you will have to insert a system diskette (boot diskette) into drive A:. Use the first MS-DOS installation diskette.

```
FORMAT driveid /S
```

The following is the command necessary for preparing a diskette in disk drive A: as a system diskette:

```
FORMAT A: /S
```

☞ If the diskette has a lower capacity than the disk drive you're using, you must specify the correct capacity using the /F: switch. To format a double density diskette for storing data and the system files on a 1.2 Meg disk drive, type:

```
FORMAT A: /S /F:360
```

This diskette can be used to boot your computer. Simply insert the diskette in drive A: and switch on your computer.

This process may take slightly longer than it did when MS-DOS 5.0 was installed on a hard drive. This is caused by the many adjustments, made by the CONFIG.SYS and AUTOEXEC.BAT files, that automatically occur when the computer is booted. So you should create both of these files on your new system diskette and also copy the driver program and other MS-DOS commands onto your system diskette.

☞ If a diskette that isn't formatted as a system diskette is in drive A: when you start the computer, MS-DOS displays an error message and prompts you to insert a system diskette and press any key. If you accidentally left a diskette in the disk drive and want to start from the hard drive, simply remove this diskette and press any key. Otherwise, exchange the data diskette for a system diskette.

Volume label and volume serial number of a diskette

After a diskette is formatted, MS-DOS 5.0 prompts you to give the diskette a volume label. This is an "electronic label". For MS-DOS Versions 3.3 and earlier, the /V switch was used to specify the volume label of a diskette during formatting:

```
FORMAT driveid /V
```

Beginning with MS-DOS 4.0, this option is no longer needed. You can ignore the volume labeling by pressing Enter.

☞ You can also use the LABEL command at any time to give a volume label to a diskette. The name of the diskette is displayed at the beginning of the directory when you use the DIR command.

The volume label you choose for your diskette can have a maximum of 11 characters and can (unlike filenames) contain spaces. You should use this method of "electronic diskette labelling" to give your diskettes meaningful names. So instead of displaying the individual files stored on the diskette, you can use a simple command (VOL) to determine what kind of information the diskette contains.

MS-DOS 5.0 doesn't use the volume labels of diskettes to distinguish between different diskettes. Instead, a volume serial number, which is automatically placed on a diskette when it's formatted, is used. This number consists of 8 hexadecimal digits. MS-DOS can use this volume serial number to determine when you've changed diskettes.

MS-DOS displays the volume label, volume serial number and other information after formatting a diskette. The DIR command also displays this information:

```
Volume in drive A is WORKDISK
Volume number: 3120-11E0
Directory of A:\
```

In this example the volume label is WORKDISK and the volume serial number is 3120-11E0.

☞ Prior to MS-DOS 3.3, a diskette did not contain a volume serial number. So when MS-DOS 5.0 displays the contents of a diskette formatted with a version of DOS earlier than DOS 3.3, the line with the volume serial number is missing. For example:

```
Volume in drive A has no label
Directory of A:\
```

Special formatting switches

You can use /1 (only 360K disk drive) to format only one side of a diskette (180K), /8 to format a diskette with 8 instead of 9 sectors and /B to format a diskette with room reserved for the operating system (boot) files. Although users seldom use these options, they are available for compatibility with earlier versions of DOS.

The following table provides an overview of the various disk drives and capacities. It also provides the specifications used with the FORMAT command:

Disk drive	Capacity	Tracks	Sectors	FORMAT: Switch
5 1/4 inch	160K	40	8	/1/8 (only PC-drive)
	180K	40	9	/1 (only PC-drive)
	320K	40	8	/8 (only PC-drive)
	360K	40	9	(on AT: /F:360)
	1.2 Meg	80	15	(only AT)
3 1/2 inch	720K	80	9	(1.44 Meg: /F:720)
	1.44 Meg	80	18	(only AT)
	2.88 Meg	80	36	(new disk drives)

Different capacities and parameters possible when using FORMAT

Information about a formatted diskette

After you finish formatting and entering a name for the diskette, information about the storage device is displayed:

```
C:\>format a: /s
Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format
Formatting 1.2M
Format complete
System transferred

Volume label (11 characters, ENTER for none)? WORK DISK

    1213952 bytes total disk space
     117760 bytes used by system
    1096192 bytes available on disk

         512 bytes in each allocation unit
        2141 allocation units available on disk

Volume Serial Number is 2928-12DD

Format another (Y/N)?
```

Diskette information after formatting

These lines have the following meanings:

total disk space:
Total storage capacity of diskette.

used by system:
Space taken up by system files (if present).

in bad sectors:
Space not available because of defects (if present).

bytes available on disk:
Space available to store data.

bytes in each allocation unit:
Smallest storage unit on the diskette; even if a file only contains one character, it takes up at least the space of one allocation unit.

allocation units available on disk:
Number of available allocation units.

volume serial number:
MS-DOS labels each new diskette with a serial number so that it can determine when you change diskettes.

☞ If the error message “Parameters not supported by drive" appears, the parameters you specified cannot be used on this disk drive (e.g., /1 on ATs or 3 1/2 inch disk drives).

☞ If the error message "Invalid medium or track 0 defective" appears, you either tried to format a diskette at a capacity too high for it (1.2 Meg) or you cannot format the system track. In the first case, use /4 for 360K. In the second case, use the diskette only for data (do not use /S).

Automatic formatting

There are a few situations in which MS-DOS automatically formats a diskette without requiring you to use the FORMAT command:

- When you use the DISKCOPY command to copy a diskette, MS-DOS automatically formats the diskette if you haven't already done so. Except for the volume serial number, the duplicate copy is identical to the source diskette. Therefore, MS-DOS automatically formats it with the correct capacity. If this isn't possible (because the source diskette was formatted for 1.2 Meg and the target diskette isn't high density (HD)), MS-DOS aborts the DISKCOPY command with an error message.

- When you use the BACKUP command to copy data, MS-DOS also automatically formats diskettes if you haven't already done so. MS-DOS uses the highest possible capacity. If you prefer to have the diskettes formatted with a lower capacity, use a special switch for BACKUP (/F). After this switch, specify the same capacities as you do for the FORMAT command. To use double density (360K) diskettes on an AT disk drive with 1.2 Meg, specify /F:360.

3.4 Formatting Hard Drives

Just as a floppy diskette must be formatted, so must a hard drive be prepared before it can be used to store data. Preparing the hard drive consists of dividing the hard drive into one or more areas (partitions) and formatting.

☞ Many PC's are sold with preformatted and partitioned hard drives. In this case, you won't have to make any changes to the partitions. Also, if you have already worked with the hard drive, you do not have to prepare it for MS-DOS. The only time you need to do this is when you want to change the division of your hard drive into one or more partitions.

Dividing the hard drive into partitions

Since hard drives can hold much more data than diskettes, you can use a hard drive for more than one operating system. In order to allow for multiple operating systems and because there are different ways of managing a hard drive, MS-DOS lets you divide a hard drive into multiple partitions.

A partition is an arbitrary number of related tracks on the hard drive. For example, on a 400 track hard drive, tracks 0 to 300 might represent one partition and tracks 301 to 400 a second partition. Partitioning allows for different operating systems to use separate parts of the hard drive and manage their data in different ways.

You can divide a hard drive into a total of four partitions. Therefore it's possible to have up to four different operating systems on one hard drive. The hard drive partitioning information is stored in the first sector of the hard drive, which is called the partition sector.

With MS-DOS Version 3.2 and earlier, a hard drive could contain only a single partition no larger than 32 Meg (a partition that DOS could use). To use a hard drive with 40 Meg capacity with MS-DOS Version 3.2 and earlier, a user was only able to use 32 Meg; the other 8 Meg were inaccessible to MS-DOS.

Starting with MS-DOS 3.3, two partitions could be defined on a hard drive, with the first or primary partition having a maximum capacity of 32 Meg. Even with an additional 32 Meg for the second, or extended partition, the DOS maximum manageable hard drive capacity was limited to 64 Meg. But 64 Meg was too limiting for some users. For this reason, it's now possible to subdivide the second, extended partition even further, into logical disk drives.

A logical disk drive can contain up to 32 Meg of data. You can add as many logical disk drives as there are identifiers (drive letters) available for disk drives (depending on the total capacity of the hard drive). Since MS-DOS supports a total of 26 disk drive identifiers from A: to Z:, a hard drive can have a maximum of 26 logical disk drives (25 + primary partition), which enables it to manage 832 Meg of hard drive capacity. Since the first two disk drive letters are usually reserved for disk drives A: and B:, practically speaking there are 24 hard drives possible, yielding a maximum capacity of 768 Meg.

Starting with MS-DOS 4.0, a partition (or logical disk drive) is no longer limited to 32 Meg. This partition can now be up to 2048 Meg (2 Gigabytes). MS-DOS 5.0 can also work with several smaller partitions.

To be able to work with MS-DOS, you must have at least a primary DOS partition installed on the hard drive. You can also install an extended DOS partition. MS-DOS must be started from the primary DOS partition. To do this, the partition must contain the operating system. So, when you format the hard drive, specify the /S switch. We'll see how to do this shortly.

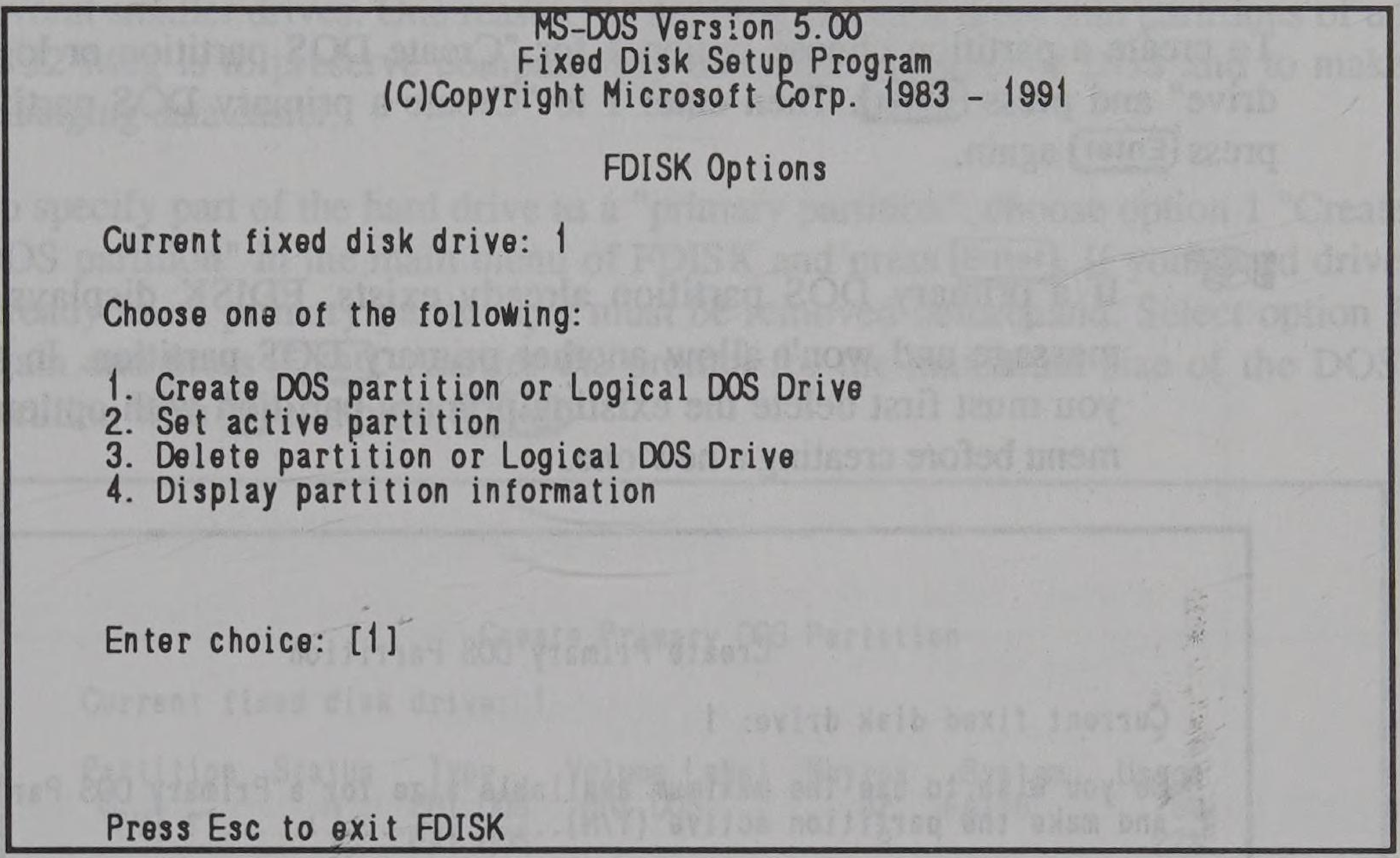
Although MS-DOS 5.0 can manage almost any size hard drive without dividing it into partitions, you may want to consider multiple partitions under the following conditions:

- If you occasionally work with another version of DOS, such as MS-DOS 3.3. Previous to MS-DOS 3.3, the maximum size for a partition was 32 Meg.
- If you want to use another operating system instead of MS-DOS (e.g. UNIX or OS/2) you must divide the hard drive into more than one partition. First install MS-DOS on one part of your hard drive and prepare the other part with the command that corresponds to the other operating system.
- If the directory or the File Allocation Table (FAT) is destroyed, you lose only the data in the partition that was destroyed. This is another good reason to divide a large hard drive into several smaller partitions.

A hard drive can only have one "primary DOS partition". If the entire capacity for this partition isn't used, you can only use the rest under MS-DOS for storing data if it is defined as an extended DOS partition. Then you'll define "logical" disk drives in the extended DOS partition. In other words, you must assign disk drive identifiers to the logical drives in the extended DOS partition.

☞ If your hard drive is already formatted and partitioned, don't run the FDISK program. You can lose all the data on your hard drive by creating new partitions on it. Only run FDISK when you need to create partitions on your hard drive. See the section in this chapter titled "Changing the partitioning" for more information.

To divide the hard drive into partitions use the FDISK command. A menu with four or five options is displayed. Option 5 is only displayed if more than one hard drive is installed in your computer.



```
                        MS-DOS Version 5.00
                      Fixed Disk Setup Program
               (C)Copyright Microsoft Corp. 1983 - 1991

                            FDISK Options

Current fixed disk drive: 1

Choose one of the following:

1. Create DOS partition or Logical DOS Drive
2. Set active partition
3. Delete partition or Logical DOS Drive
4. Display partition information


Enter choice: [1]




Press Esc to exit FDISK
```

Display of FDISK menu

To select an option, type the number corresponding to the option and press Enter. Press Esc to return to the previous menu. In the main menu, pressing Esc terminates the FDISK command. If you've already made changes to the partition data, reboot the computer with an MS-DOS system diskette.

☞ If you use FDISK to make changes to the hard drive partitions and your hard drive is already installed and formatted, you should write down the names of all of the logical disk drives. FDISK prompts you to enter the logical drive letter before it deletes each logical disk drive.

You can specify the size of a partition either in megabytes or percentages. If you enter a numerical value, FDISK interprets this as a value in megabytes. To enter a percentage, type a percentage symbol (%) after the number.

➡ All of the data on a hard drive will be lost when partitions are changed.

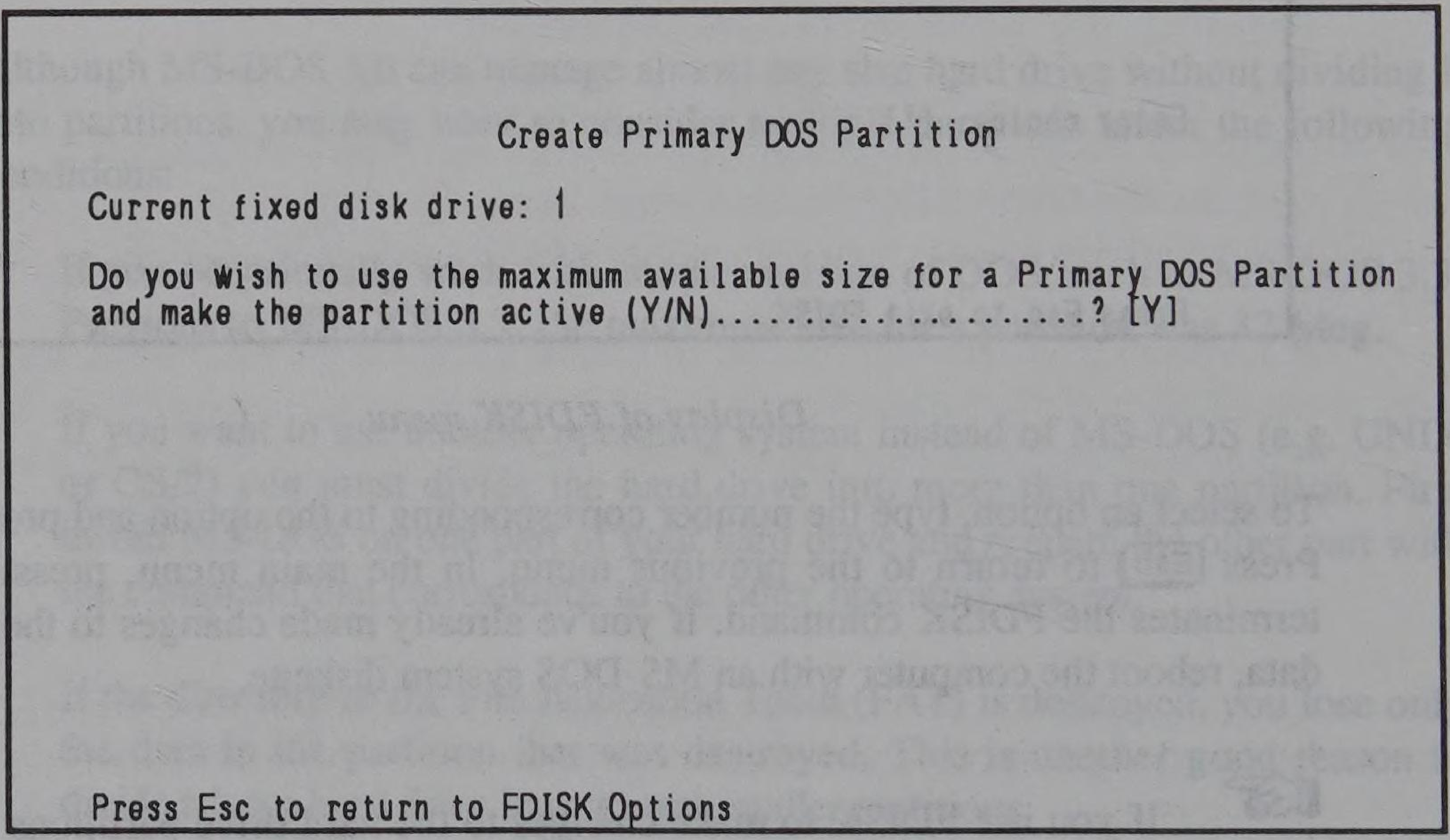
Creating a primary DOS partition for the entire hard drive

The easiest way to prepare a hard drive for working with MS-DOS is to create a single partition for starting the computer (primary partition).

Changing partitions with FDISK (even when you're creating a primary DOS partition) deletes all of the data on the hard drive.

To create a partition choose option 1 for "Create DOS partition or logical disk drive" and press Enter. Then enter 1 to "Create a primary DOS partition" and press Enter again.

If a primary DOS partition already exists, FDISK displays an error message and won't allow another primary DOS partition. In this case, you must first delete the existing primary partition with option 3 of the menu before creating a new one.



```
                       Create Primary DOS Partition

Current fixed disk drive: 1

Do you wish to use the maximum available size for a Primary DOS Partition
and make the partition active (Y/N)....................? [Y]









Press Esc to return to FDISK Options
```

Create primary DOS partition

To the above prompt, answer by typing Y and pressing Enter. This causes MS-DOS to treat the entire hard drive as a single partition. Since no further partitions are possible, FDISK ends.

Reboot after partitioning

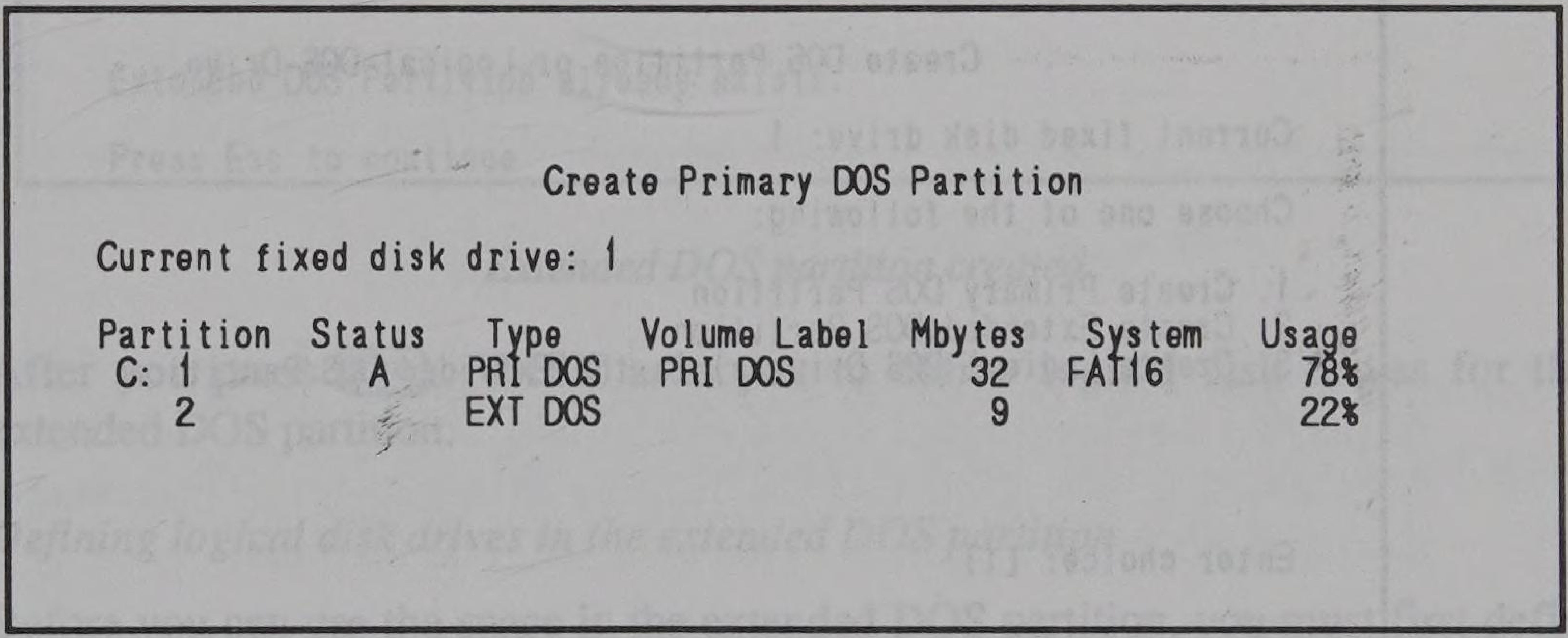
Insert the first installation diskette in the disk drive and press Enter. The computer then automatically reboots and continues the installation program.

You still won't be able to store any data on a hard drive prepared in this way. The hard drive must first be formatted. MS-DOS determines if the drive has been formatted during installation and automatically formats the hard drive if necessary.

Creating a primary DOS partition (partitioning the hard drive)

MS-DOS 5.0 can use hard drives with practically unlimited capacity as a single partition identified by a single drive identifier. But it's usually best to divide it into several smaller drives. One reason for dividing the hard drive into partitions of up to 32 Meg is to preserve compatibility to earlier versions of DOS and to make managing data easier.

To specify part of the hard drive as a "primary partition", choose option 1 "Create DOS partition" in the main menu of FDISK and press Enter. If your hard drive already has a primary partition, it must be removed beforehand. Select option 1 again and press Enter. Answer the prompt for the maximum size of the DOS partition with N and press Enter.



```
                    Create Primary DOS Partition

Current fixed disk drive: 1

Partition  Status   Type    Volume Label  Mbytes  System  Usage
 C: 1        A     PRI DOS   PRI DOS        32    FAT16    78%
    2              EXT DOS                   9             22%
```

Specifying a portion of the hard drive as a primary partition

The screen displays the total size and available size in percentages. The actual size displayed depends on your particular hard drive. Enter the desired size in megabytes (numbers only) or in percentages (numbers followed by percentages). Any errors can be corrected by pressing the Backspace key and retyping the corrections. Press Enter when your entry is complete.

FDISK then displays the new partition and the corresponding statistical information. First the disk drive letter is displayed; the next number is the partition number. This is followed by the type of partition (PRI DOS for primary DOS partitions and EXT DOS for extended DOS partitions). The volume label and size of the partition then follow.

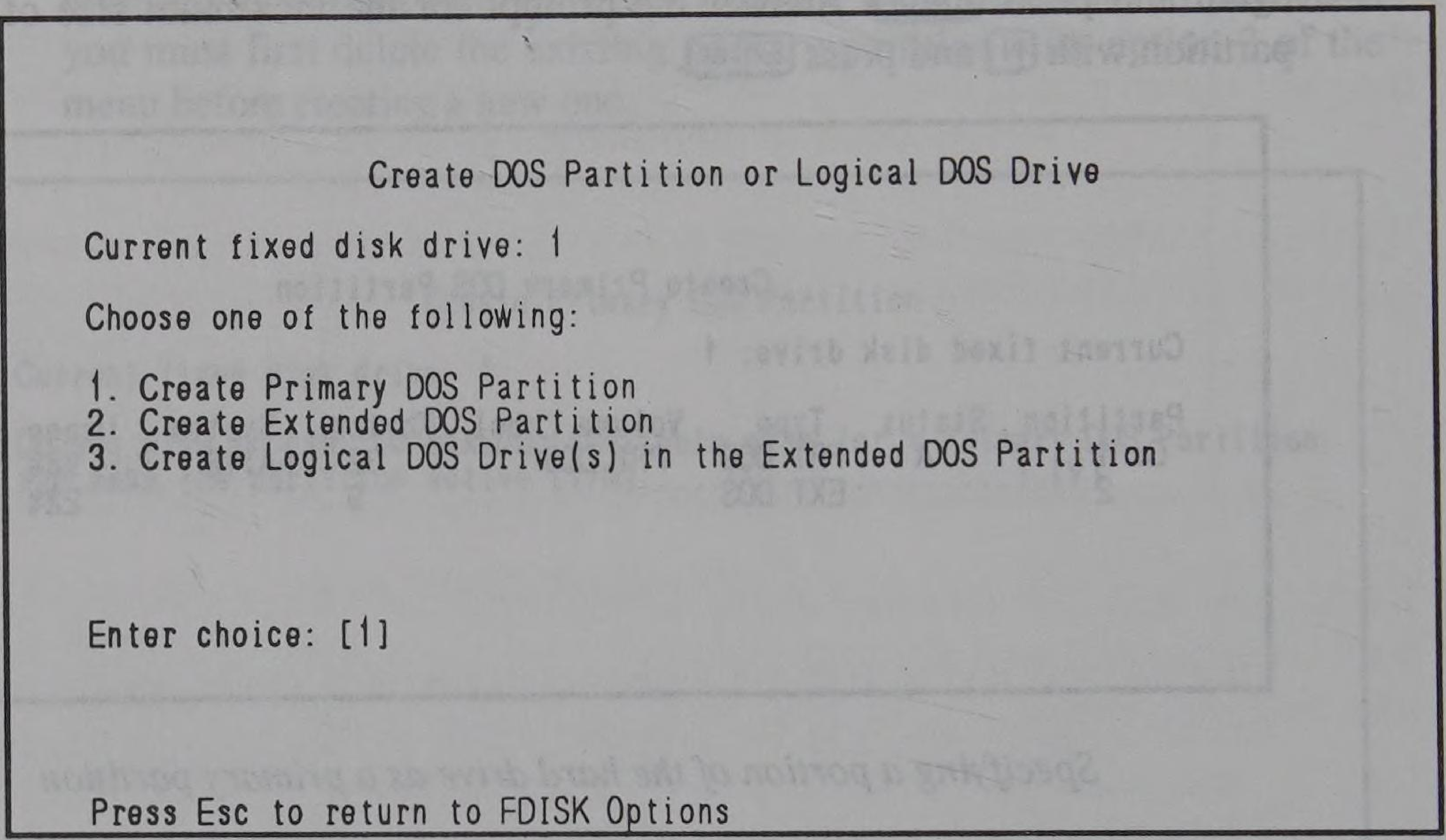
After creating the primary partition, you can create extended DOS partitions. Press Esc to return to the main menu of FDISK.

You can't boot from the hard drive when you create a primary DOS partition. You must first specify which partition is to be the active partition by using option 2 of the menu.

Creating an extended DOS partition

If you create a primary DOS partition and don't specify the total capacity of the hard drive, you can define an extended DOS partition. If you don't define an extended DOS partition, you won't be able to use the rest of the hard drive. You must create "logical" disk drives in the partition before the extended DOS partition is assigned a disk drive letter.

To define an extended DOS partition, enter 1 in the main menu for "Create DOS partition or Logical DOS Drive" and press Enter. Then select 2 for "Create extended DOS partition".

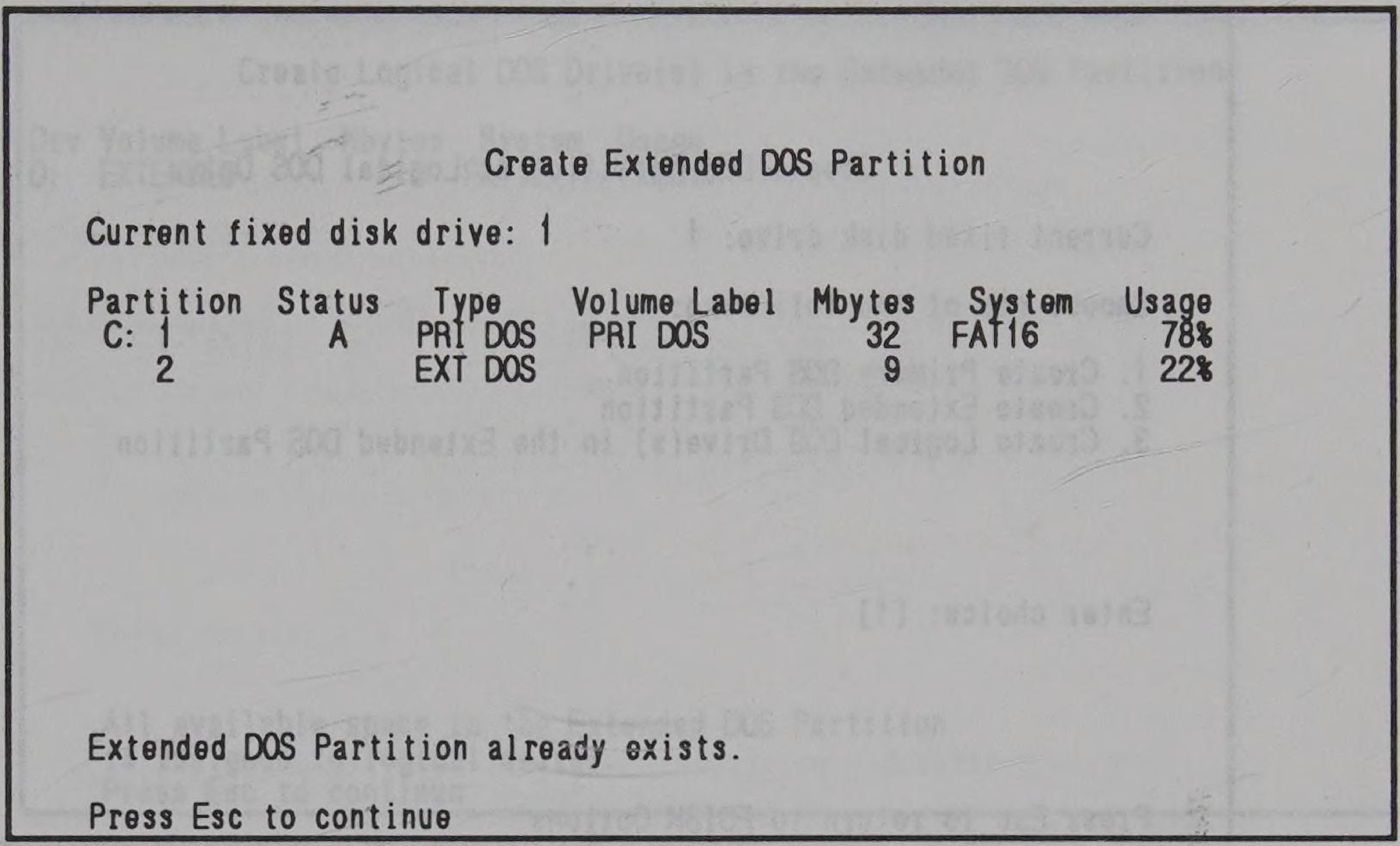


```
                    Create DOS Partition or Logical DOS Drive

   Current fixed disk drive: 1

   Choose one of the following:

   1. Create Primary DOS Partition
   2. Create Extended DOS Partition
   3. Create Logical DOS Drive(s) in the Extended DOS Partition




   Enter choice: [1]




   Press Esc to return to FDISK Options
```

Select 2 to create an extended DOS partition

FDISK displays the current status of the hard drive with the defined primary DOS partition. If you want to use all of the hard drive for MS-DOS, assign the rest of the hard drive to the extended DOS partition. Otherwise, the rest of the hard drive can't be used in DOS because MS-DOS cannot manage more than two partitions (one primary and one extended). If you want to use another operating system on the hard drive (UNIX, OS/2) in addition to MS-DOS, leave enough space on the hard drive to run the other operating system.

You can enter the size of the extended DOS partition either in megabytes or as a percentage. FDISK automatically calculates the rest of the hard drive for the extended DOS partition. Use the Backspace key to delete any mistakes in your entry. Press Enter after completing your entry; FDISK displays the message "Extended DOS partition created". The screen displays the current division of the hard drive:



```
                      Create Extended DOS Partition

Current fixed disk drive: 1

Partition  Status   Type    Volume Label  Mbytes   System   Usage
 C: 1        A    PRI DOS   PRI DOS          32    FAT16      78%
    2             EXT DOS                     9               22%





Extended DOS Partition already exists.

Press Esc to continue
```

Extended DOS partition created

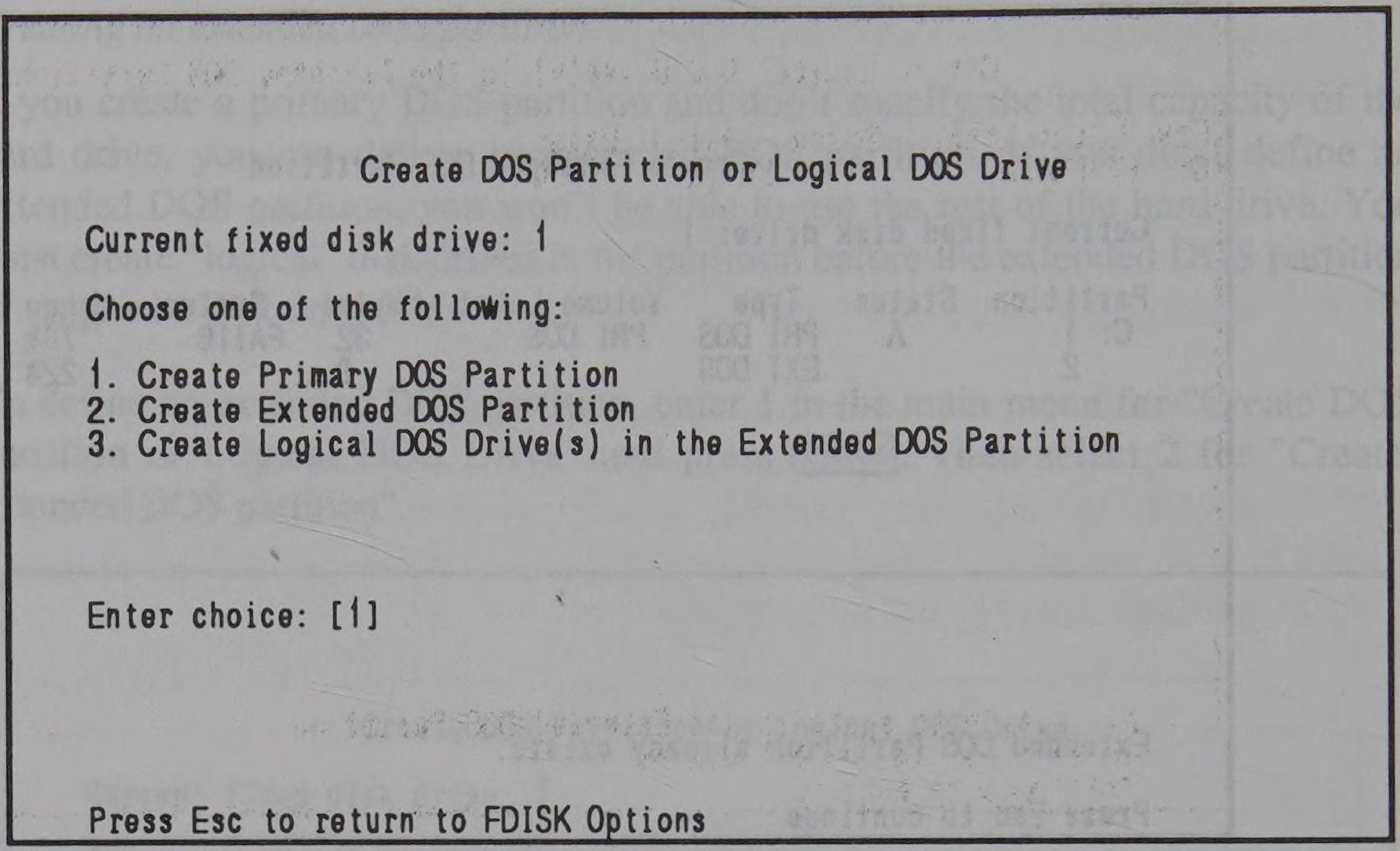
After you press Esc, FDISK asks you to define logical disk drives for the extended DOS partition.

Defining logical disk drives in the extended DOS partition

Before you can use the space in the extended DOS partition, you must first define one or more logical disk drives. To do this, choose option 1 "Create DOS partition" in the FDISK main menu. Then choose option 3 to define logical disk drives.

When you create an extended DOS partition, FDISK automatically activates this menu option.

The screen displays the total size of the extended DOS partition and the amount available in megabytes. Instead of the entire hard drive, the percentage only refers to the size of the extended partition. If there aren't any existing logical disk drives, the screen displays a message (showing 100% available capacity) that tells you this.



```
                    Create DOS Partition or Logical DOS Drive

     Current fixed disk drive: 1

     Choose one of the following:

     1. Create Primary DOS Partition
     2. Create Extended DOS Partition
     3. Create Logical DOS Drive(s) in the Extended DOS Partition



     Enter choice: [1]




     Press Esc to return to FDISK Options
```

Defining logical disk drives

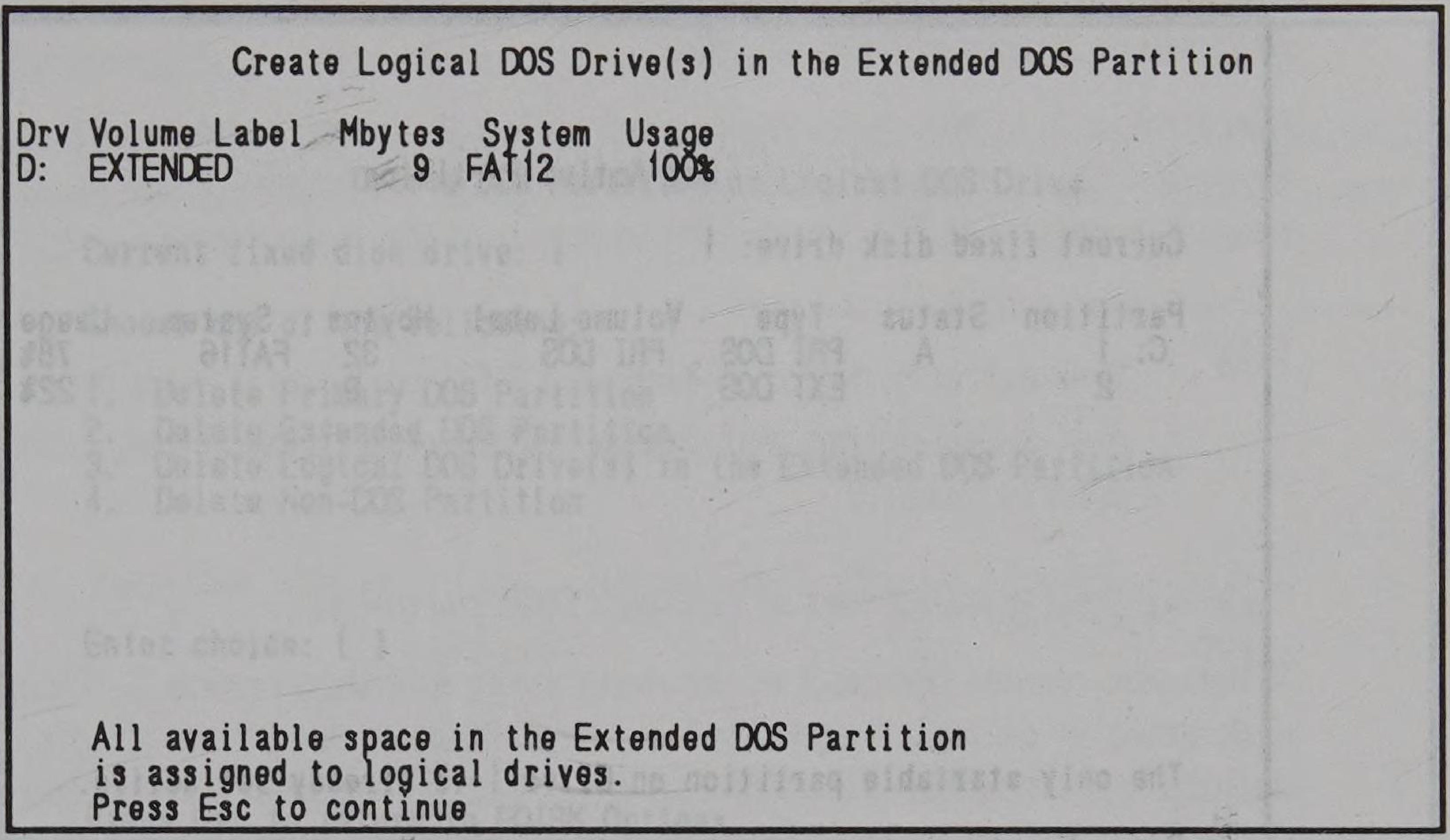
If there are already logical disk drives defined, FDISK will let you know by displaying the identifiers of these logical drives. Now you can enter the size for the logical disk drives, specifying the capacity in megabytes or percentages. FDISK then displays the new division of the extended DOS partition.

The program assigns the next unused letter for each newly defined logical disk drive. As long as the total capacity of the extended DOS partition has not been assigned to logical disk drives, FDISK will continue to prompt you to define more disk drives.

While you can terminate this process by pressing Esc, the rest of the hard drive would not be available to DOS.

☞ If you want to retain the ability for DOS 3.3 and earlier to access the data in one of these partitions, you must assign only a maximum capacity of 32 Meg for the logical disk drives.

If the total capacity has already been allocated, FDISK displays the current division. Use Esc to return to the main menu.



```
           Create Logical DOS Drive(s) in the Extended DOS Partition

Drv Volume Label  Mbytes  System  Usage
D:  EXTENDED           9  FAT12    100%










     All available space in the Extended DOS Partition
     is assigned to logical drives.
     Press Esc to continue
```

Logical disk drive in the extended DOS partition

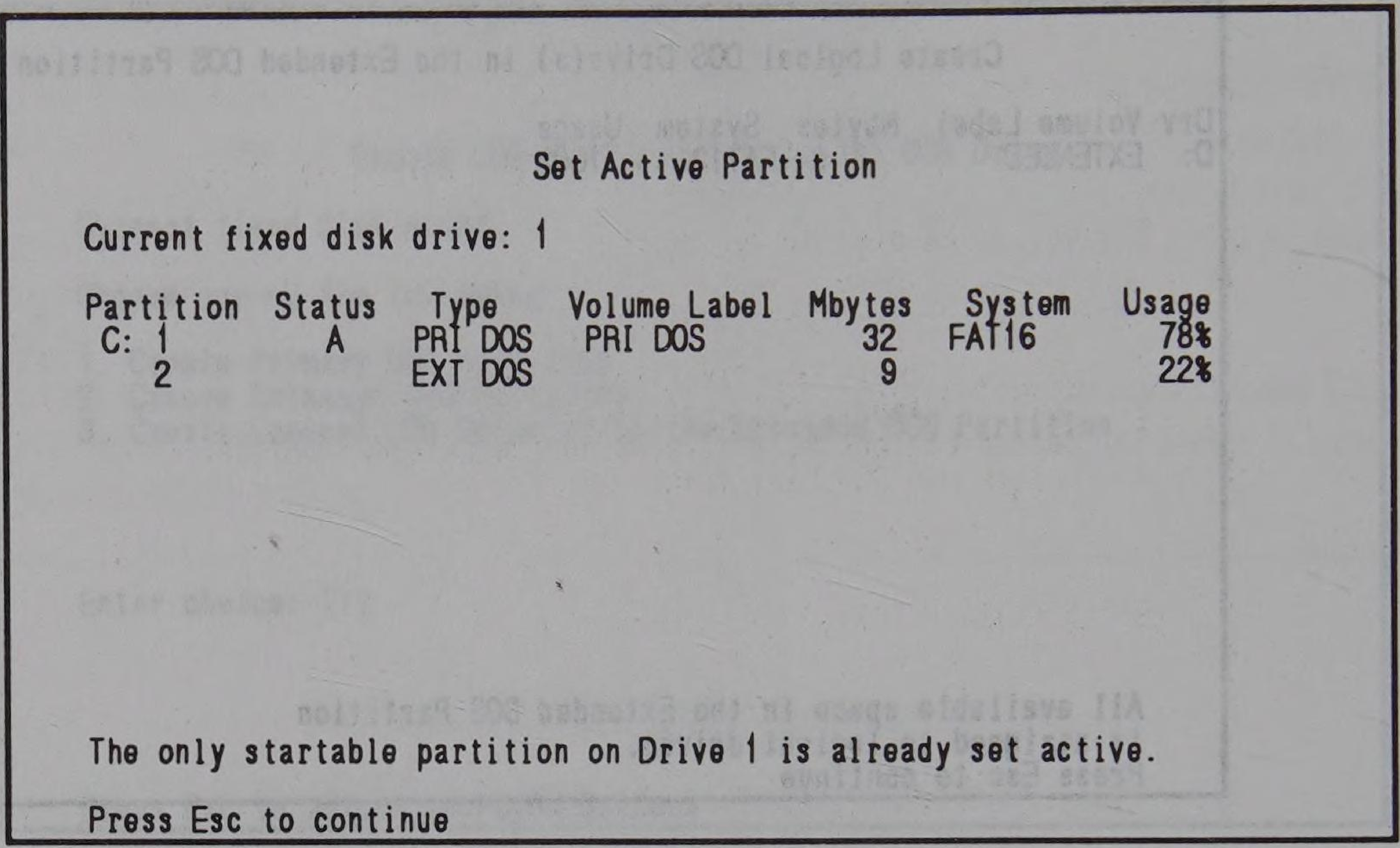
Logical disk drives are sections of the hard drive that you can address using their own disk drive identifiers. When you create logical disk drives, you're dividing your hard drive into multiple disk drives.

Remember the disk drive letters of these logical disk drives. You'll use these disk drive letters later to access the different parts of the hard drive.

Setting the active partition

Use menu option 2 to specify the active DOS partition. Start MS-DOS from this partition, just as you would from a system diskette.

If there isn't an active partition, FDISK informs you of this and indicates that you cannot boot the computer from the hard drive.



```
                       Set Active Partition

Current fixed disk drive: 1

Partition  Status   Type    Volume Label  Mbytes   System   Usage
 C: 1         A    PRI DOS   PRI DOS          32   FAT16      78%
    2              EXT DOS                     9              22%





The only startable partition on Drive 1 is already set active.

Press Esc to continue
```

Setting the active partition

After choosing this option, you must enter the number of the desired partition and press Enter. If a hard drive only has MS-DOS partitions, you can only select the primary DOS partition. After you set the active partition, the program displays the current status of the hard drive. Return to the main menu by pressing Esc.

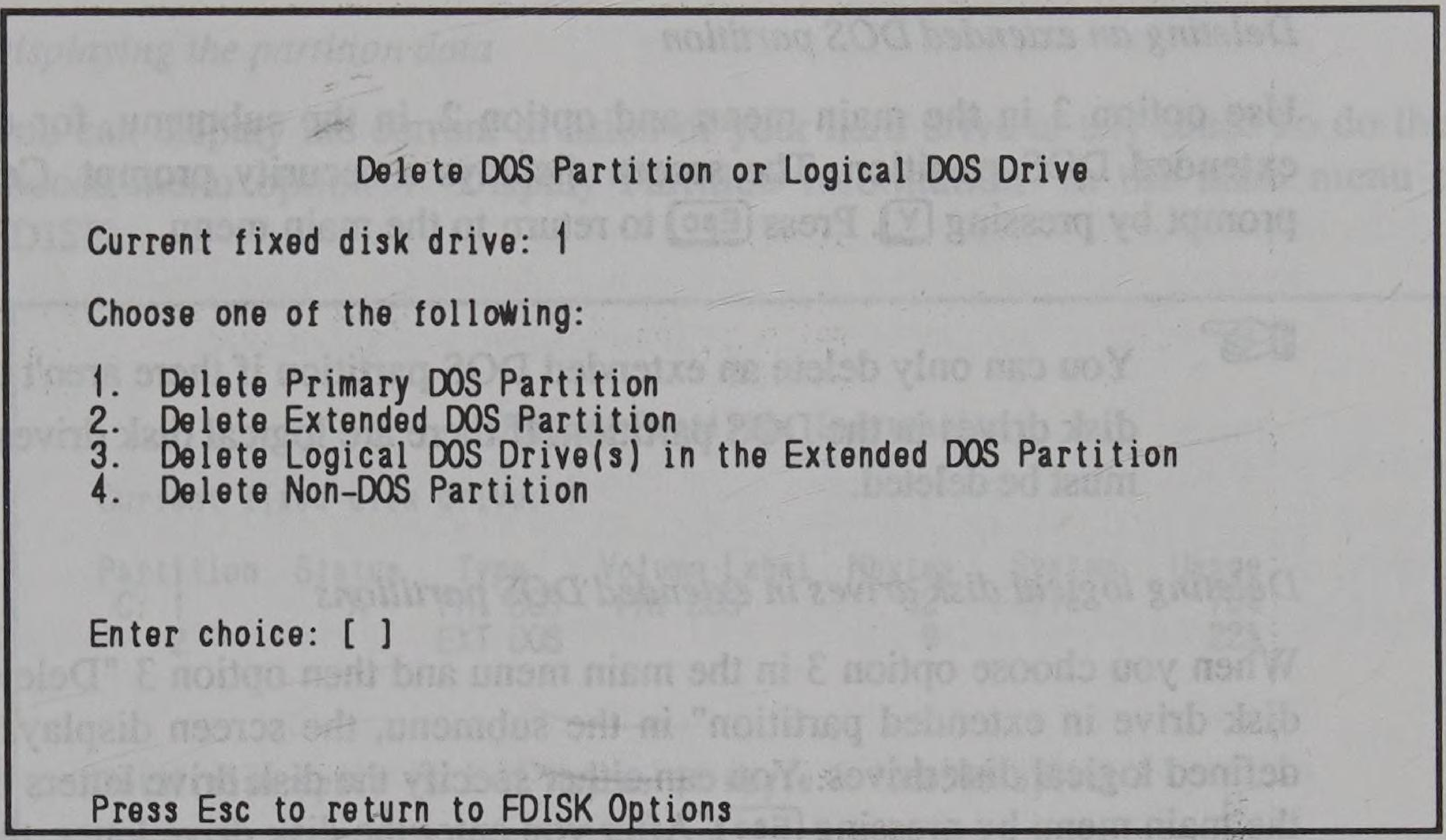
☞ If the entire hard drive is set up for MS-DOS or if there is only one DOS partition defined as the active partition, FDISK displays an error message. Return to the main menu by pressing Esc.

Deleting a primary DOS partition

When you delete the primary DOS partition, the data on the partition will be lost.

☞ You can only delete a primary DOS partition if there is no longer an extended DOS partition. In order to be deleted, the primary DOS partition must be the last remaining partition.

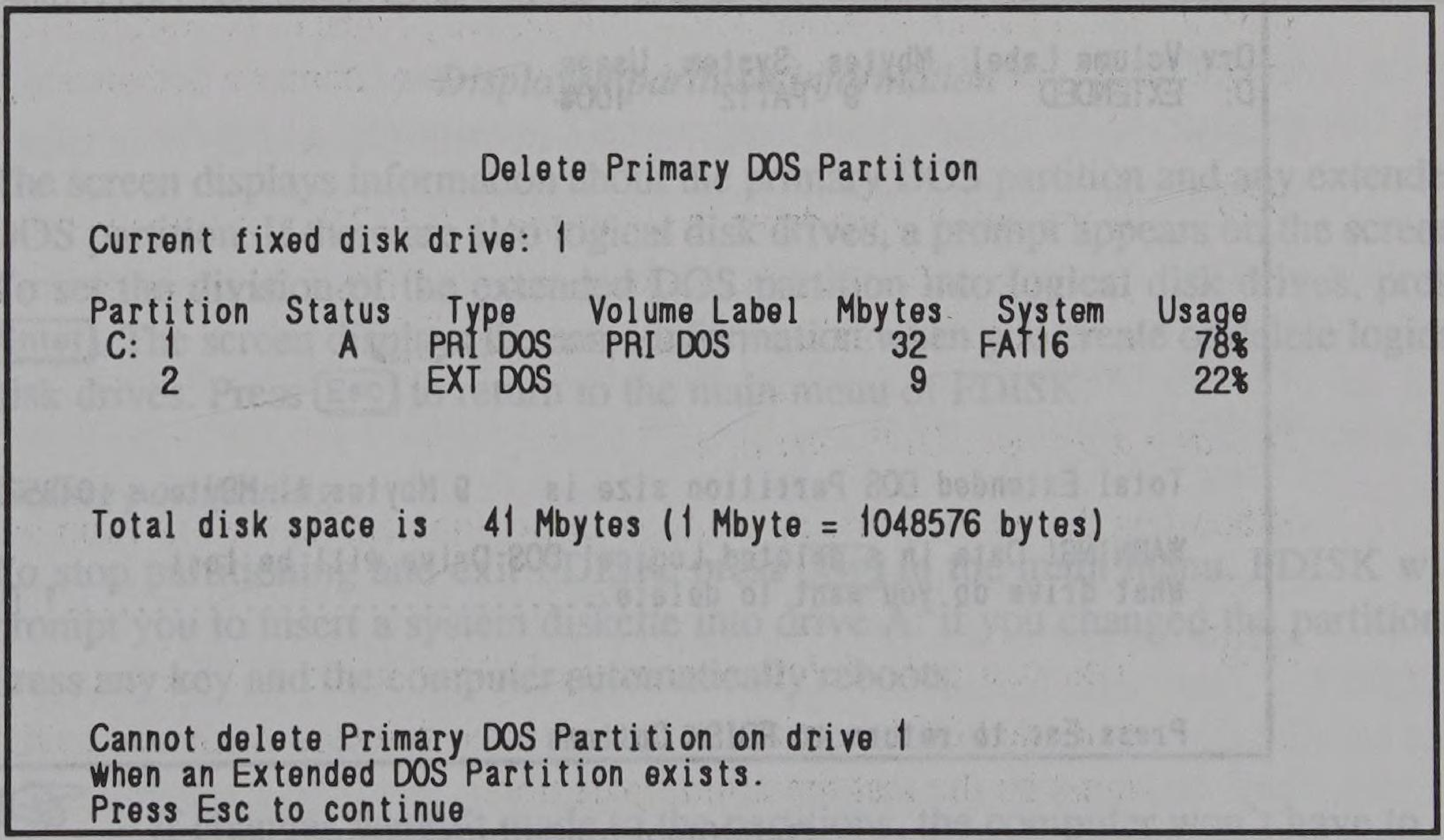
Select option 3 (Delete DOS partition or logical disk drive) in the FDISK main menu. Then activate option one, in the submenu, for deleting a primary DOS partition. A security prompt is displayed. Confirm by pressing Y.



```
                Delete DOS Partition or Logical DOS Drive

Current fixed disk drive: 1

Choose one of the following:

1.  Delete Primary DOS Partition
2.  Delete Extended DOS Partition
3.  Delete Logical DOS Drive(s) in the Extended DOS Partition
4.  Delete Non-DOS Partition


Enter choice: [ ]



Press Esc to return to FDISK Options
```

Deleting the primary DOS partition

The message "Primary DOS partition deleted" appears and the screen displays the new status of the hard drive.



```
                    Delete Primary DOS Partition

Current fixed disk drive: 1

Partition  Status   Type    Volume Label  Mbytes   System   Usage
 C: 1         A    PRI DOS   PRI DOS        32     FAT16      78%
    2              EXT DOS                   9                22%



Total disk space is   41 Mbytes (1 Mbyte = 1048576 bytes)





Cannot delete Primary DOS Partition on drive 1
when an Extended DOS Partition exists.
Press Esc to continue
```

Primary DOS partition deleted

Press Esc to return to the main menu of FDISK. Be careful when you press Esc because pressing this key too many times aborts FDISK. If this happens you must reboot the computer.

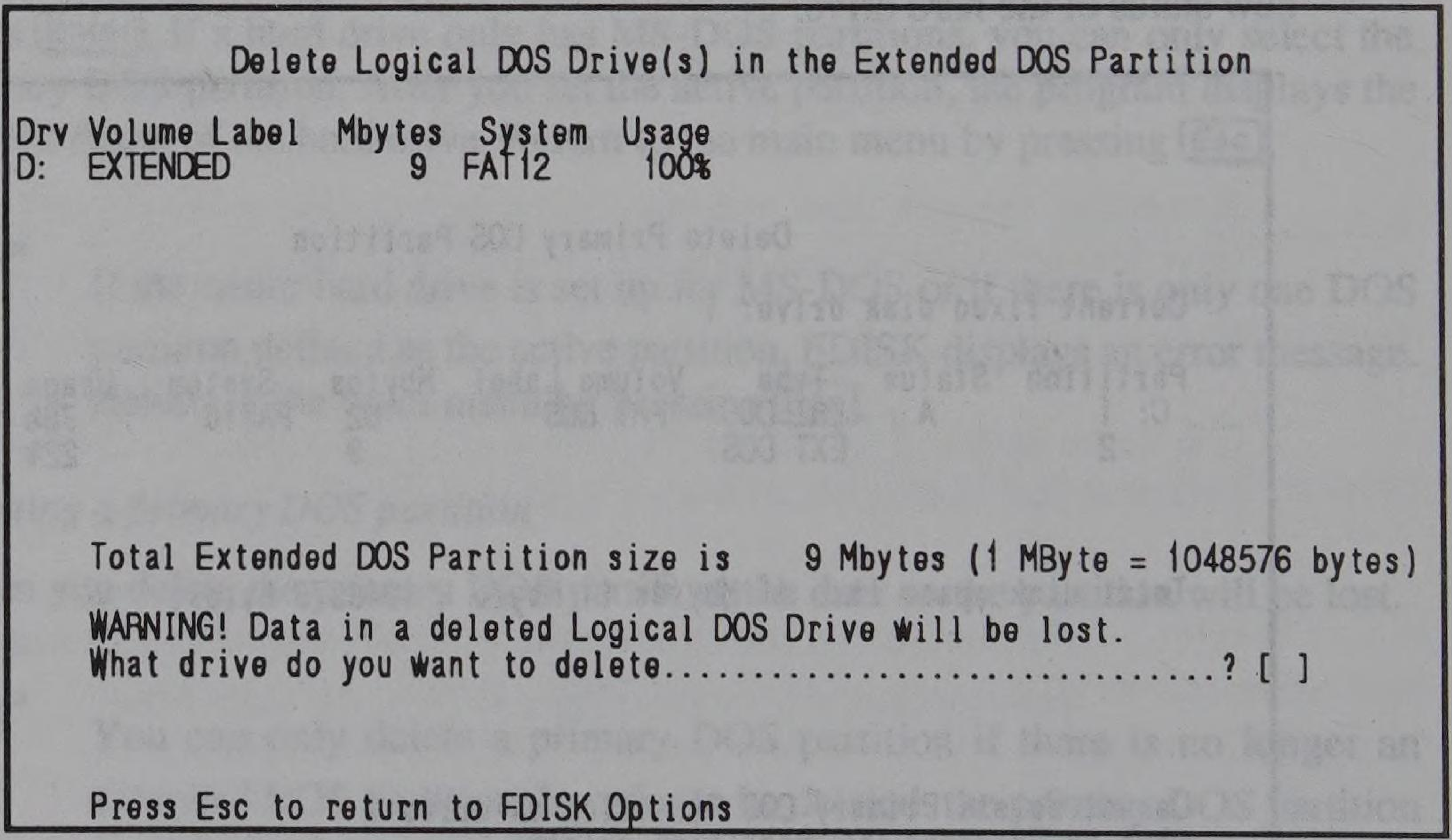
Deleting an extended DOS partition

Use option 3 in the main menu and option 2, in the submenu, for deleting an extended DOS partition. The screen displays a security prompt. Confirm this prompt by pressing Y. Press Esc to return to the main menu.

☞ You can only delete an extended DOS partition if there aren't any logical disk drives in the DOS partition. If there are logical disk drives, first they must be deleted.

Deleting logical disk drives in extended DOS partitions

When you choose option 3 in the main menu and then option 3 "Delete a logical disk drive in extended partition" in the submenu, the screen displays all of the defined logical disk drives. You can either specify the disk drive letters or return to the main menu by pressing Esc. After you enter the disk drive letter, the program prompts you for the volume label. Enter the volume name, which can be set with the MS-DOS VOL command. If the disk drive doesn't have a name, simply press Enter.



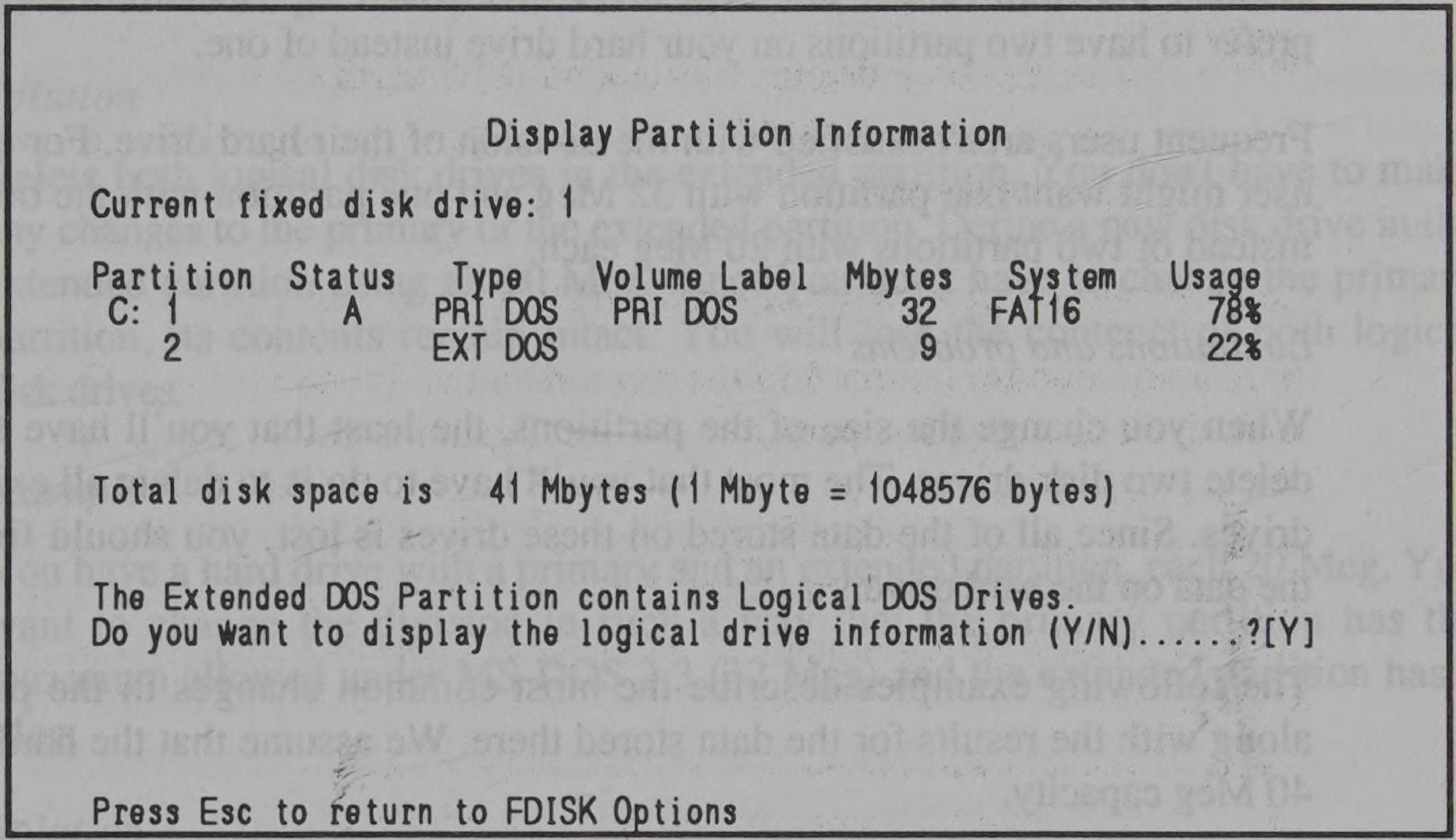
```
            Delete Logical DOS Drive(s) in the Extended DOS Partition

Drv Volume Label  Mbytes  System  Usage
D:  EXTENDED          9   FAT12    100%








    Total Extended DOS Partition size is    9 Mbytes (1 MByte = 1048576 bytes)

    WARNING! Data in a deleted Logical DOS Drive will be lost.
    What drive do you want to delete.............................? [ ]



    Press Esc to return to FDISK Options
```

Deleting a logical disk drive

Following the security prompt, the disk drive definition is marked "deleted" in the display on the screen. After you delete the last logical disk drive in the extended DOS partition, the screen displays a message informing you that there aren't any defined logical disk drives. You cannot use the capacity of the extended DOS partition. Use Esc to return to the main menu.

Displaying the partition data

You can display the current division of your hard drive at any time. To do this, choose menu option 4 "Display Partition Information" in the main menu of FDISK.



```
                    Display Partition Information

Current fixed disk drive: 1

Partition  Status   Type    Volume Label  Mbytes   System    Usage
 C: 1         A    PRI DOS   PRI DOS        32     FAT16      78%
    2              EXT DOS                   9                22%



Total disk space is   41 Mbytes (1 Mbyte = 1048576 bytes)


The Extended DOS Partition contains Logical DOS Drives.
Do you want to display the logical drive information (Y/N).....?[Y]




Press Esc to return to FDISK Options
```

Displayed partition information

The screen displays information about the primary DOS partition and any extended DOS partition. If there are also logical disk drives, a prompt appears on the screen. To set the division of the extended DOS partition into logical disk drives, press Enter. The screen displays the same information when you create or delete logical disk drives. Press Esc to return to the main menu of FDISK.

Ending partitioning

To stop partitioning and exit FDISK, press Esc in the main menu. FDISK will prompt you to insert a system diskette into drive A: if you changed the partitions. Press any key and the computer automatically reboots.

☞ If changes weren't made to the partitions, the computer won't have to be rebooted. Press Esc; the command interpreter's system prompt appears. This happens if you were simply looking at the partition information.

Changing the partitioning

Although the dealer usually makes the first division of your hard drive, you may often need to change the partitioning. In the following section we'll provide examples and reasons for changing the partitioning.

Reasons for changing the partitioning

There are several reasons for changing the partitioning of a hard drive. For example, you could have used MS-DOS 3.3 first and divided your 40 Meg hard drive into two partitions. MS-DOS 5.0 lets you now use the entire hard drive as a primary partition (under one disk drive identifier). Or you might find that you prefer to have two partitions on your hard drive instead of one.

Frequent users aren't satisfied with the division of their hard drive. For example, a user might want one partition with 32 Meg and one partition with the other 8 Meg instead of two partitions with 20 Meg each.

Limitations and problems

When you change the size of the partitions, the least that you'll have to do is to delete two disk drives. The most that you'll have to do is to delete all existing disk drives. Since all of the data stored on these drives is lost, you should first backup the data on the hard drive.

The following examples describe the most common changes in the partitioning along with the results for the data stored there. We assume that the hard drive has 40 Meg capacity.

Example 1

You originally divided the hard drive into two partitions of 20 Meg each. Now you want to use the entire hard drive as one partition.

Solution

Both partitions must be deleted and a new primary partition must be created with the total capacity. By doing this, all of the data on the entire hard drive is lost. Using FDISK, first delete the logical disk drive in the extended DOS partition, then delete the partition. Now you can remove the primary DOS partition. Finally, choose 1 for creating a primary DOS partition twice, answer each prompt for the maximum size of the DOS partition by pressing Y and Enter.

You'll need to reboot the computer. Since you destroyed the contents of the entire hard drive, your computer will no longer boot from the hard drive. Place a boot diskette that contains MS-DOS 5.0 into the disk drive. Now you'll need to reformat the hard drive. You should also copy the system onto the hard drive (using the FORMAT command with the /S switch).

Example 2

You have a hard drive with a primary partition (20 Meg) and an extended partition (also 20 Meg). The extended partition has two logical disk drives with 10 Meg each. You want to combine both of these logical disk drives into one logical disk drive with 20 Meg.

Solution

Delete both logical disk drives in the extended partition. You don't have to make any changes to the primary or the extended partition. Define a new disk drive in the extended partition using all 20 Meg. Since you don't have to change the primary partition, its contents remain intact. You will lose the contents of both logical disk drives.

Example 3

You have a hard drive with a primary and an extended partition, each 20 Meg. You want to change the division in such a way that the primary partition has the maximum allowed under MS-DOS 3.3 (32 Meg) and the extended partition has 8 Meg.

Solution

Remove the extended and the primary partition by deleting the logical disk drives in the extended DOS partition. Then remove the extended DOS partition and then the primary DOS partition.

Define a new primary partition with 32 Meg and an extended partition with the remaining 8 Meg. Now create the logical disk drive in the extended partition. Deleting both partitions causes all of the data on the hard drive to be lost. Reformat both disk drives (primary partition and logical disk drive). Format the primary partition with the system (FORMAT /S).

Formatting the hard drive

You can format a hard drive with the FORMAT command after you use FDISK to divide it into one or more partitions. FORMAT identifies any defects on the hard drive and marks these areas so that they're not used later. FORMAT also deletes all existing data on the hard drive.

So if you're going to take your computer to the repair shop and want to remove sensitive data from the hard drive, you should format the hard drive. Or perhaps a software error has caused serious defects in the structure of the data on the drive and it can no longer be repaired with CHKDSK.

If the hard drive is divided into several partitions and logical disk drives, you must format each individually using the FORMAT command. When formatting, specify the disk drive letter. Format the primary disk drive with the /S switch to copy the system files from the system diskette to the hard drive. You can then boot your computer from this disk drive.

On earlier versions of DOS you needed to know the volume label to format the hard drive, this is not required in the current version of MD-DOS 5.0, hopefully it will be added to later versions. A security prompt will be issued before the format process begins, only answer Y if you really want to format the hard drive. If you accidently format your hard disk you can usually recover it with the UNFORMAT command.

If you format a hard drive with MS-DOS 4.0 or higher, MS-DOS 5.0 will display the volume serial number and the volume label.

☞ You can also use the MS-DOS Shell to format your hard drive with the group of Disk Utilities programs. Start FORMAT. You can then enter any parameters for the FORMAT command. The following section describes these parameters. Be careful: If you try to format the hard drive with the operating system from the MS-DOS Shell, you will definitely have problems. This process deletes all of the files of the MS-DOS Shell and all MS-DOS commands in general.

Formatting the hard drive with system files

When you format the primary DOS partition, you should transfer the system files so that you can boot your PC from the hard drive. The system files are the hidden files IO.SYS and MSDOS.SYS and the command interpreter COMMAND.COM on your disk. If you don't transfer the system files when you format drive C:, you won't be able to boot the computer from the hard drive. Use the /S switch to transfer the system files to your hard drive:

```
FORMAT driveid /S Enter
```

After typing this, if the operating system (including the hidden files) still isn't on the hard drive and there isn't a system diskette with these files in drive A:, the FORMAT command prompts you to place a DOS diskette in drive A: and confirm by pressing Enter.

☞ If you try to format the hard drive in DOS 5.0, a security prompt will be issued before the format process begins, only answer Y if you really want to format the hard drive. If you accidently start to format your hard drive you should let the formatting process finish. Don't shut off or reboot the computer! You can usually recover the data on the hard drive with the UNFORMAT command as long as the UNFORMAT command is issued as soon as the formatting process is finished.

After you enter the FORMAT command, the screen displays a security prompt to warn you that you're about to erase all of the data in the drive. Only confirm your entry by pressing Y if you are sure you wish to format the hard drive. The program counts off a percentage from 0 to 100. When the formatting is completed, specify a volume label for the hard drive. The volume label can have a maximum of 11 characters and can contain spaces. On earlier versions of DOS you needed to know the volume label to format the hard drive, this may not be required in DOS 5.0. If you choose not to give your hard drive a volume label, press Enter to skip the label.

The screen displays the total memory and available memory:

total disk space:	Total storage capacity of the hard drive or partition.
used by system:	Occupied by system files.
in bad sectors:	Space not available because of defects (if present).
bytes available on disk:	Space available to store data.

☞ Since the hard drive contains the operating system, it uses a part (approximately 110K) of the hard drive. MS-DOS often indicates this as: "in hidden files".

Formatting other hard drive partitions

If the hard drive contains an extended DOS partition with logical disk drives, you must also FORMAT them:

```
FORMAT driveid
```

☞ If you accidently start to format a hard drive you should let the formatting process finish. Don't shut off or reboot the computer! You can usually recover the data on the hard drive with the UNFORMAT command. Run the UNFORMAT command as soon as the formatting process is finished.

A prompt warns you that you're about to erase all of the data on the drive. A security prompt will be issued before the format process begins, only answer Y if you really want to format the hard drive. The screen displays the progress of FORMAT as a percentage. When FORMAT is completed, enter a volume name for the drive. (Press Enter to skip this option.) The screen then displays the total storage capacity for the drive.

☞ Since you can only start MS-DOS from the primary DOS partition, you don't have to transfer the system files to these other logical drives. The screen doesn't display "used by system".

In earlier versions of DOS formatting a diskette caused the data to be destroyed immediately. DOS 5.0 uses the MIRROR and UNFORMAT commands to protect your data disks from accidental formatting. Before a formatted disk is formatted, UNFORMAT information is saved on the disk. If you press Ctrl + Alt + Del during the formatting process you may lose some of your data. It is better to press Ctrl + C to abort the formatting process. You could even let the formatting process complete itself. You can then usually recover the data on the hard drive with the UNFORMAT command as long as the UNFORMAT command is issued as soon as the formatting process is finished.

➡ This may not apply to some versions of DOS that are customized for hardware manufacturers. We recommend that you don't try formatting your disks to check these commands.

MS-DOS 5.0 automatically gives the drive a volume serial number, which is displayed by several other DOS commands (VOL, DIR, CHKDSK, etc.).

3.5 Labeling Storage Media

MS-DOS allows you to assign a volume label your storage media. Many DOS commands identify the media by its volume label.

☞ If a hard drive is divided into several partitions and logical disk drives, you can give each partition/logical disk drive a different volume label.

Labeling a storage media

Use the LABEL command to give a volume label to a storage media:

```
LABEL driveid
```

The screen displays the previous volume label of the storage media. You can enter a new volume label with a maximum of 11 characters. Unlike filenames, a volume label name can contain imbedded spaces. However, the following characters cannot be used:

<	less than
>	greater than
+	plus
=	equal
,	comma
;	semicolon
:	colon
.	period
*	asterisk
?	question mark
(	open parenthesis
)	close parenthesis
[	open bracket
]	close bracket
/	slash
\	backslash

If you don't specify a driveid, LABEL uses the current disk drive. When you enter the LABEL command you can specify the new volume label of the storage device directly after the colon:

```
LABEL A:vollabel
```

Deleting the name of a storage media

To delete a volume label, use the LABEL command again but press Enter without typing a new label. Answer the prompt that follows by pressing Y.

Asking for names

You can use the VOL command at any time to determine the volume label of a storage media:

```
VOL
```

If you don't specify a disk drive, VOL automatically displays the volume label of the current disk drive. To determine the volume label of the diskette in drive A:, use the command:

```
VOL A:
```

In addition to the volume label, the screen also displays a volume serial number consisting of eight (two pair of four hexadecimal) digits. MS-DOS 5.0 automatically assigns this volume serial number when the media is formatted. Even if two storage devices have the same name, this number is used to identify a storage device:

```
VOL C:

Volume in disk drive C: is DR_C
Volume Serial Number is 3817-13CC
```

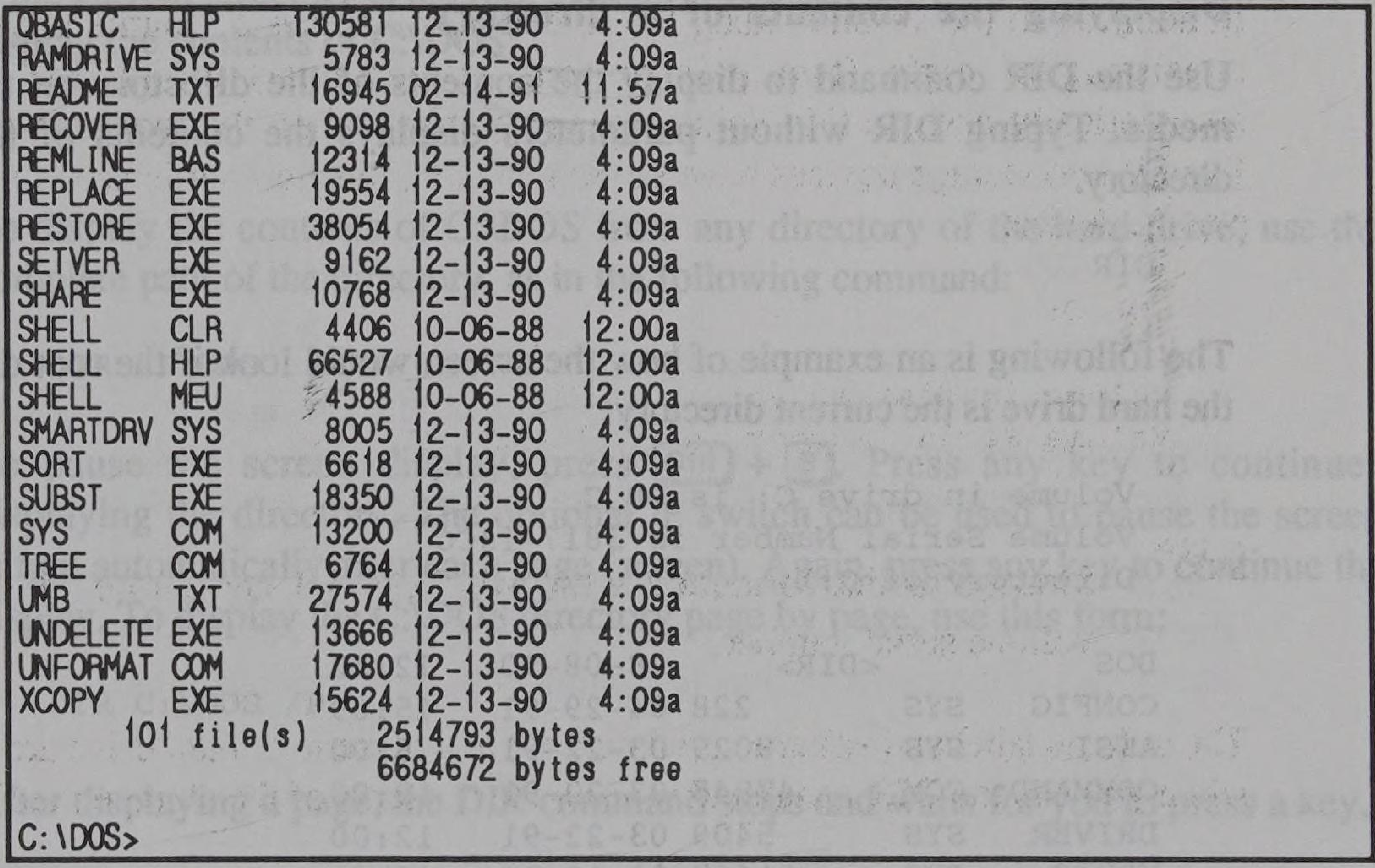
☞ Storage media formatted with MS-DOS 3.3 and earlier don't have a volume serial number.

The volume label and volume serial number of a diskette or hard drive are displayed when you use the DIR command to view the contents of the storage media.

3.6 Displaying The Contents

A storage media can contain files and/or directories. A file is a group of data, such as a letter or spreadsheet information. Files appear in the disk directory as separate entries, which contain the filename, extension, file size and the date the file was created. Files contain stored data.

A subdirectory is an arbitrary collection of files that can be identified by a separate name. A directory collects the names of these separate files under a single name. By removing the individual filenames from the main directory (root directory) a SUB directory helps maintain order in the storage device.



```
QBASIC    HLP   130581 12-13-90   4:09a
RAMDRIVE SYS      5783 12-13-90   4:09a
README    TXT    16945 02-14-91  11:57a
RECOVER   EXE     9098 12-13-90   4:09a
REMLINE   BAS    12314 12-13-90   4:09a
REPLACE   EXE    19554 12-13-90   4:09a
RESTORE   EXE    38054 12-13-90   4:09a
SETVER    EXE     9162 12-13-90   4:09a
SHARE     EXE    10768 12-13-90   4:09a
SHELL     CLR     4406 10-06-88  12:00a
SHELL     HLP    66527 10-06-88  12:00a
SHELL     MEU     4588 10-06-88  12:00a
SMARTDRV SYS      8005 12-13-90   4:09a
SORT      EXE     6618 12-13-90   4:09a
SUBST     EXE    18350 12-13-90   4:09a
SYS       COM    13200 12-13-90   4:09a
TREE      COM     6764 12-13-90   4:09a
UMB       TXT    27574 12-13-90   4:09a
UNDELETE EXE     13666 12-13-90   4:09a
UNFORMAT COM     17680 12-13-90   4:09a
XCOPY     EXE    15624 12-13-90   4:09a
       101 file(s)    2514793 bytes
                      6684672 bytes free

C:\DOS>
```

Disk directory

Setting the current storage device

Many MS-DOS commands affect the current storage media or the current disk drive, unless otherwise specified. To activate another disk drive, enter the disk drive identifier (letter plus colon):

```
X:
```

A: and B: are usually disk drives and C: represents the first hard drive. To make your hard drive the current drive, type the letter followed by a colon:

```
C:Enter
```

Use the PROMPT command to automatically display the current drive in a system prompt:

```
PROMPT $N
```

We recommend using "PG" as your system prompt because it causes the screen to display the current drive and current directory:

```
PROMPT $P$G
```

If your hard drive (C:) is the current drive, the system prompt would look like the following:

```
C:\>
```

Displaying the contents of a directory

Use the DIR command to display the contents of the directory on the storage media. Typing DIR without parameters displays the contents of the current directory.

```
DIR
```

The following is an example of how the screen would look if the root directory of the hard drive is the current directory:

```
 Volume in drive C: is DR_C
 Volume Serial Number is 3817-13CC
 Directory of C:\

DOS          <DIR>      08-08-90   12:42
CONFIG   SYS        228 04-29-91   15:09
ANSI     SYS       9029 03-22-91    5:00
COMMAND  COM      47845 03-22-91   12:00
DRIVER   SYS       5409 03-22-91   12:00
MOUSE    SYS      31687 12-12-89   16:49
CONFIG   BAK        245 08-07-90   14:25
AUTOEXEC BAK        224 08-07-90   14:25
AUTOEXEC BAT        221 08-07-90   15:10
TEXTS        <DIR>      08-08-90   13:00
TOOLS        <DIR>      08-08-90   13:00
WORD         <DIR>      08-08-90   13:14
        12 File(s)      78134 bytes
                        7190528 bytes free
```

The beginning of the display shows general information about the storage device (volume label, volume serial number and the name of directory being displayed). Next, a list of existing files and subdirectories in the root directory is displayed. You can identify a file because there will be a number representing the file size in front of the date.

Subdirectories are indicated by the characters <DIR>. In the second to last line, DIR displays the total number of files and subdirectories and the amount of storage space occupied by the files, and in the last line the amount of unused storage space available on the drive.

To display the contents of a subdirectory, specify the name of the directory and, if necessary, the complete path:

```
DIR directoryname
```

This form of the DIR command has the same meaning as:

```
DIR directoryname\*.*
```

If the current directory is the root directory, you can use the following command to display the contents of C:\DOS:

```
DIR DOS
```

To display the contents of C:\DOS from any directory of the hard drive, use the complete path of the directory, as in the following command:

```
DIR C:\DOS
```

To pause the screen display, press Ctrl + S. Press any key to continue displaying the directory. The optional /P switch can be used to pause the screen output automatically after each page (screen). Again, press any key to continue the display. To display the C:\DOS directory page by page, use this form:

```
DIR C:\DOS /P
```

After displaying a page, the DIR command stops and waits for you to press a key.

Displaying all of the directories of a storage media

Use the TREE command to display all of the directories of a storage media. This command displays only the directories; files aren't displayed. The output appears in a tree structure:

```
Directory PATH listing for Volume DR_C
Volume Serial Number is 3817-13CC
C:.
├───COMP
├───BAT
├───MSDOS
├───ORG
├───TEMP
├───TEXTS
│   ├───COMPANY
│   └───PRIVATE
├───DATA
│   ├───1986
│   ├───1987
│   ├───1988
│   ├───1989
│   └───1990
├───WORD
│   └───SECURITY
├───CURRENT
└───DOS
```

The TREE command presents a graphic picture of the organization of a disk drive by directory. This option is especially valuable for changing directories when you don't know the path of the directory to which you want to change. In our example, to see the contents of the 1986 directory, first move to the Data directory.

The deeper a directory is within the directory structure, the farther it is indented to the right. If you don't specify a directory after the TREE command, all of the directories of the current drive, beginning with the current directory, are displayed:

```
TREE
```

If C:\DATA is the current directory, TREE displays only this directory and its subdirectories:

```
Directory PATH listing for Volume DR_C
Volume Serial Number is 3817-13CC
C:\DATA
├───1986
├───1987
├───1988
├───1989
└───1990
```

To display all of the directories of the current disk drive, specify this as follows:

```
TREE driveid\
```

For example, if drive A: is the current disk drive and you want to display all of the directories on drive C:, use the following command:

```
TREE C:\
```

Displaying all of the directories and files of the storage device

You can display all of the directories and files of a storage device by adding the /F switch to the TREE command. If you don't enter a drive identifier after TREE, all of the directories and files, beginning with the current drive, will be displayed:

```
TREE /F
```

To display all of the directories and files of any disk drive, specify the root directory of the drive:

```
TREE driveid\ /F
```

TREE then displays the directory name within the tree structure and the files of the directory below it:

```
Directory PATH listing for Volume DR_C
Volume Serial Number is 3817-13CC
C:.
|   CONFIG.SYS
|   ANSI.SYS
|   COMMAND.COM
|   DRIVER.SYS
|   MOUSE.SYS
|   CONFIG.BAK
|   AUTOEXEC.BAK
|   AUTOEXEC.BAT
|
├───COM
|      MEYER01.DOC
|      MEYER02.DOC
|      MEYER01.BAK
|      MEYER02.BAK
|      INVOICE.STY
```

Since there is so much information, more than one screen page may be needed to display the entire list. You can pause the display after each page by using the MORE filter. To continue the display, press any key.

For example, to display all of the files on drive C: page by page, use the following command:

```
TREE C: /F | MORE
```

The pipe symbol (|) links two commands together: the TREE and MORE commands. If a pipe symbol isn't on your keyboard, hold down the Alt key and enter the digits 1 2 4 on the numeric keypad (to the right of the keyboard).

If you use the MORE filter, it takes a few seconds before the files are displayed to the screen.

3.7 Copying Diskettes

DISKCOPY can duplicate the entire contents of a diskette to another diskette. The duplicate will have the same format as the source diskette. When the disk is copied, any data on the target diskette is lost. So it's a good idea to write protect the source diskette so that you don't accidently lose the data by inserting the source diskette into the disk drive at the wrong time. Since DISKCOPY requires you to swap the source and target disk several times, it's easy to lose track of which disk should be in the drive at any given time.

Use DIR to examine the directory of the target diskette before you duplicate the diskette so that you can be sure that you don't overwrite important data.

If the target diskette hasn't been formatted, DISKCOPY automatically formats it for you. After you insert the target diskette, a message informs you of the formatting process. DISKCOPY takes longer using an unformatted target disk.

☞ The target diskette is automatically formatted with the same storage capacity as the source diskette. However, the computer must be capable of doing this. You cannot copy an HD diskette with 1.2 Meg onto a DD diskette. DISKCOPY will realize that it cannot attain the desired capacity on the target diskette and abort with an error message. Also the disk drives must be of the same type. You cannot copy a 5 1/4 inch diskette onto a 3 1/2 inch diskette.

After you insert the source diskette, DISKCOPY displays a message about the diskette's format:

```
Copying 80 tracks
15 sectors per track, 2 side(s)
```

In the above example, the source diskette is a 5 1/4 inch diskette with 1.2 Meg capacity.

Unlike the COPY and XCOPY commands, DISKCOPY creates an identical copy of the source diskette. So instead of just the data, you're also transferring the relative location of the data on the diskette. This means that the data on the copy is arranged in the same way as the source diskette.

That's why DISKCOPY is the best command to use for making backup copies of irreplaceable diskettes. For example, you should use DISKCOPY to make backup copies of all of the diskettes included with a program.

☞ In contrast to earlier versions of DOS, source and target diskettes aren't 100 percent identical. DISKCOPY automatically gives the target diskette a new volume serial number. In spite of this, MS-DOS 5.0 treats both diskettes as though they were identical; the DISKCOMP command for comparing diskettes won't find any differences in MS-DOS 5.0. Comparisons in other versions of DOS will indicate a difference on side 0, track 0, since this is where the volume serial number is stored.

You can only make a copy if the target diskette is free of defects. If the target diskette cannot be formatted completely and correctly, the copying process aborts after the message "Formatting While Copying" appears. The following error message appears:

```
Drive types or diskette types
not compatible
```

In this case, use a different diskette for your target diskette and retry the operation. Although you could continue using the defective diskette by formatting it with FORMAT, a message, informing you that there are X number of bytes in defective sectors, appears. These defective sectors are automatically excluded from data storage.

DISKCOPY always copies the entire diskette, regardless of the amount of data stored on it. The process takes just as long when the diskette is completely blank (but formatted). You cannot copy an unformatted diskette. If you're copying only a few small files it might be better to use the COPY command instead of DISKCOPY.

Since DISKCOPY produces a duplicate of the source diskette, it also preserves the relative location of data on the diskette when copying. Even though this can be an advantage at times, it's usually a disadvantage because the longer you work with a diskette, the worse the distribution of the data becomes. When this happens the data becomes fragmented. MS-DOS can process data faster when it is stored contiguously. Copying the files with COPY or XCOPY onto a blank diskette provides an ideal arrangement of your files. So it's better to copy data by files instead of using DISKCOPY whenever possible.

Copying with a single disk drive

When you copy a diskette with a single disk drive, you must insert the source and target diskettes alternately into the same disk drive. Be very careful not to mix up the source and target diskettes.

If your computer doesn't have a hard drive, you'll need to insert the MS-DOS diskette containing the DISKCOPY command into the disk drive before you can type the command. DISKCOPY is an external command:

```
DISKCOPY A: A:
```

Next, insert the source diskette into drive A: and press a key. After a few seconds, you'll be prompted to insert the target diskette into drive A: and press a key. Depending on the available main memory and capacity of the diskette, DISKCOPY will prompt you to insert the source and target diskettes into drive A: several times. When the process is finished, DISKCOPY asks whether you need to copy another diskette. Answer either by pressing Y or if you don't want to copy another diskette, press N.

If you start DISKCOPY from a disk drive, you don't have to specify the drive for the source and the target. The current drive is automatically used.

Don't start DISKCOPY from within an application program. You won't have a lot of main memory available and you'll have to swap diskettes more often. It's a lot quicker to exit the application program and reload the program after you finish copying.

Copying with two disk drives

When you copy a diskette with two disk drives, you won't have to swap the source and target diskettes. However, if the computer doesn't have a hard drive you must insert the appropriate MS-DOS diskette before using DISKCOPY so that the computer can load the DISKCOPY command:

```
DISKCOPY A: B:
```

Insert the source diskette into disk drive A: and the target diskette into disk drive B: and press any key. When the copying process is finished, DISKCOPY prompts you to copy another diskette. Press Y to confirm or N if you don't want to copy.

If the target diskette isn't formatted, the computer automatically formats it with the correct capacity.

Unfortunately, DISKCOPY doesn't support making multiple copies of the same diskette. To create two copies of one diskette, the computer must read the data from the source diskette twice. DISKCOPY also doesn't support extended memory. So even if you have 2 Meg in your PC, you will still need to swap the source diskette and the target diskette at least three times.

Verifying when you copy diskettes

If you want to be absolutely certain that the data on the copied diskette is identical and free of defects, you can specify the /V switch (for verify). To copy a diskette in drive A: and have the program automatically verify the process, use the command:

```
DISKCOPY A: A: /V
```

Since using this switch slows down the process considerably, you may choose to use it only for copying crucial data.

3.8 Comparing Diskettes

Use the DISKCOMP command to verify whether two diskettes have the same contents. The disks should have identical formats. For example, a 5 1/4 inch diskette cannot be compared with a 3 1/2 inch diskette. The command doesn't affect the contents of the diskettes. DISKCOMP doesn't simply check for the same contents, it also checks for the identical relative location of data on both diskettes. So DISKCOMP is mainly suited for checking diskettes copied by DISKCOPY. If you used COPY or XCOPY to copy the data onto another diskette, DISKCOMP will find many differences, which are caused by the rearrangement of the data. In these instances, use COMP to compare the files.

DISKCOMP can only compare diskettes with the same format and capacity. If the format is different, DISKCOMP terminates, after you insert the second diskette, with the message:

```
Drive types or diskette types
not compatible

Compare process ended

Compare another diskette (Y/N)?
```

Use COMP to compare files if the diskettes have different formats.

☞ When comparing files, DISKCOMP doesn't compare the different volume numbers. So a comparison of storage devices copied by DISKCOPY will be positive, while earlier versions of DOS (such as MS-DOS 3.3) will display messages indicating differences.

Single disk drive

Comparing is basically the same as copying a diskette with DISKCOPY. If there isn't a hard drive, you must insert the correct MS-DOS diskette before calling DISKCOMP because it is also an external command:

```
DISKCOMP A: A:
```

Next, insert the first diskette in drive A: and press any key. DISKCOMP automatically determines the format of the diskette and displays it on the screen. Depending on the amount of available main memory and the capacity of the diskette, DISKCOMP will prompt you to insert both diskettes into drive A: several times.

If both diskettes are identical, the following message appears:

```
Compare ok
```

If the diskettes aren't identical, DISKCOMP displays the differing elements found during the comparison, for example:

```
Compare error on
side 0, track 0

Compare error on
side 1, track 0
```

After the process is completed, DISKCOMP prompts you to compare more diskettes. Press Y to confirm or N if you don't need to compare anymore diskettes.

If you start DISKCOMP from a disk drive, you won't have to specify the drive. The program automatically uses the current drive.

☞ With DISKCOMP you don't have to change 1.2 Meg diskettes as often as you do with DISKCOPY. On computers with 640K main memory and typical configuration, you must insert the source and target diskette three times for DISKCOPY and only twice for DISKCOMP. The reason for this is that DISKCOPY can only read from the source and write to the target. DISKCOMP can read more data for the comparison and then compare it with the former source diskette, after comparing it with the target diskette.

Two disk drives

Comparing diskettes on a PC with two similar size disk drives is also similar to copying a diskette:

```
DISKCOMP A: B:
```

Insert the first diskette into drive A:, insert the second diskette into drive B: and press any key. DISKCOMP then displays information about the comparison and gives you the option of comparing two more diskettes.

3.9 Checking Storage Media

Checking media is an important and necessary process. If there are errors or defects on the media, you may not be able to use the data.

So if potential errors and problems on a storage media are discovered early, it's less likely that damage will occur. In this section we'll discuss the ways to check media. Use these tools at regular intervals.

CHKDSK

You can use the external command CHKDSK to check media and display information about them. This command can be used in various ways to check the structure of the data on a storage media. CHKDSK recognizes the following errors on a storage media:

- Portions of files are no longer allocated to the file.
- Files have invalid allocations.
- The same parts of the storage device are allocated to more than one file at the same time.
- Access to parts of a directory is no longer possible.
- The File Allocation Table (FAT) is defective.
- Directory is defective.
- Parts of the storage media can no longer be read correctly.

Fortunately for users, CHKDSK not only recognizes errors, but in many cases repairs them. Since attempting to repair a defective diskette can cause further changes to its contents, you must specify the /F switch. Without this switch, CHKDSK only checks for and displays errors on the screen.

Information about capacity and existing files/directories

To examine media for errors, use the following:

```
CHKDSK driveid
```

If you don't specify a drive, the contents of the current drive are examined.

If no errors are detected on the drive, CHKDSK displays information similar to the following:

```
Volume DR_C         created 04.12.1989 06.12
Volume Serial Number is 3817-13CC

  21309440 bytes total disk space
     73728 bytes in 3 hidden files
     49152 bytes in 19 directories
  13963264 bytes in 697 user files
     40960 bytes in defective sectors
   7182336 bytes available on disk

      2048 bytes in each allocation unit
     10405 allocation units on diskette/hard drive
      3507 available allocation units on diskette/hard drive

    655360 bytes main memory
    537632 bytes free
```

The items have the following meanings:

total disk space	Storage capacity of drive.
hidden files	Number of files that are hidden from most MS-DOS commands. You can only display hidden files with special auxiliary programs or the MS-DOS Shell. The system files for starting the computer from this drive are hidden files, called IO.SYS and MSDOS.SYS.
directories	Number of directories and storage space required for managing directories.
user files	Number of "normal" files and programs and storage space used.
defective sectors	Space that cannot be used due to defects.
Available on diskette/hard drive	Space still available for files and directories.

The information about the allocation units provides an explanation of the smallest storage units that MS-DOS can use to store data. 360K and 720K diskettes have 1024 bytes per allocation unit while 1.2 Meg and 1.44 Meg diskettes have only 512 bytes per allocation unit. Hard drives have allocation units containing either 2048 bytes or a multiple of 2048. A file always takes up at least one allocation unit.

CHKDSK also displays the amount of memory in the computer and the memory that is available for user programs. This information doesn't refer to the storage media, but to the computer itself.

Information about errors on the storage device

If CHKDSK finds errors on the storage device, it displays them. The program compares the File Allocation Table (FAT) to the existing files and directories. CHKDSK reports any differences as errors.

Any files on the storage media that you can no longer access with MS-DOS are indicated. Then CHKDSK displays the number of allocation units (clusters) that are affected (lost allocation units) and the number of files (chains) that can no longer be accessed. If the allocation of the clusters within a file is destroyed, CHKDSK may report more than one chain. The following is an example of what a lost allocation unit on a diskette might look like if you used CHKDSK without any additional parameters:

```
Volume WORK      created 02.03.1990 16.54
Volume Serial Number is 3916-17FC
Errors found. F parameter not specified,
Corrections will not be written to disk


   1 lost allocation unit found in 1 chain.
Convert lost chains to files (Y/N)?y

    362496 bytes total disk space
      1024 bytes in 1 user file
      1024 bytes in 1 restored file
    360448 bytes available on diskette/hard drive

      1024 bytes in each allocation unit
       354 allocation units on diskette/hard drive
       352 available allocation units on diskette/hard drive

    655360 bytes total memory
    537632 bytes free
```

At the beginning a warning message tells us that CHKDSK will not make corrections to errors it finds because you didn't specify the /F switch. Next it tells us that it has found 1 lost allocation unit in a chain. The prompt to convert the chains to files has no effect since the corrections are not written to the disk.

If you run CHKDSK and find lost allocation units, you should run it again with the /F switch. This converts the lost chains to files.

☞ Use CHKDSK at regular intervals to check important diskettes and your hard drive for potential errors.

Viewing the checking process

To get a better idea of where the error occurred, you can display the checking as it progresses:

```
CHKDSK driveid /V
```

For example, to check a diskette in drive A: and display specific information as it checks the diskette, use the command:

```
CHKDSK A: /V
```

Since the information may be extensive, you can redirect the messages from CHKDSK to a file or a printer. This produces a written error report. For example, to check all of the hard drive (C:) and output all of the information to the printer, use the following command line:

```
CHKDSK C: /V >PRN
```

Unfortunately CHKDSK doesn't have an option that lets you automatically stop screen output after every page and continue by pressing a key. To check the entire hard drive and display the information on the screen page by page, it's necessary to include the MORE command by using the following command line:

```
CHKDSK C: /V | MORE
```

☞ If your keyboard doesn't have the "|" character, you can reproduce it by holding down Alt and pressing 1, 2, 4 on the numeric keypad.

If an error appears, you'll immediately know which directory or files are affected.

Correcting errors on a storage media

When you discover errors on a storage media, you should correct them as soon as possible. However, before doing this, save all of the information on the media to prevent an even greater loss of data. If you're working with a floppy diskette, use DISKCOPY to create a backup copy. On a hard drive, use the BACKUP command. Then use CHKDSK to correct the errors.

Correcting errors in the data structure

To correct errors on the storage device, assign the /F switch to CHKDSK. You can choose to convert lost allocation units to files or have CHKDSK release them for further use.

In the first case, CHKDSK creates a file in the root directory named FILE????.CHK, where ???? is a consecutive number, beginning with 0000. If the lost data are ASCII files, you can display them with TYPE and possibly rebuild them.

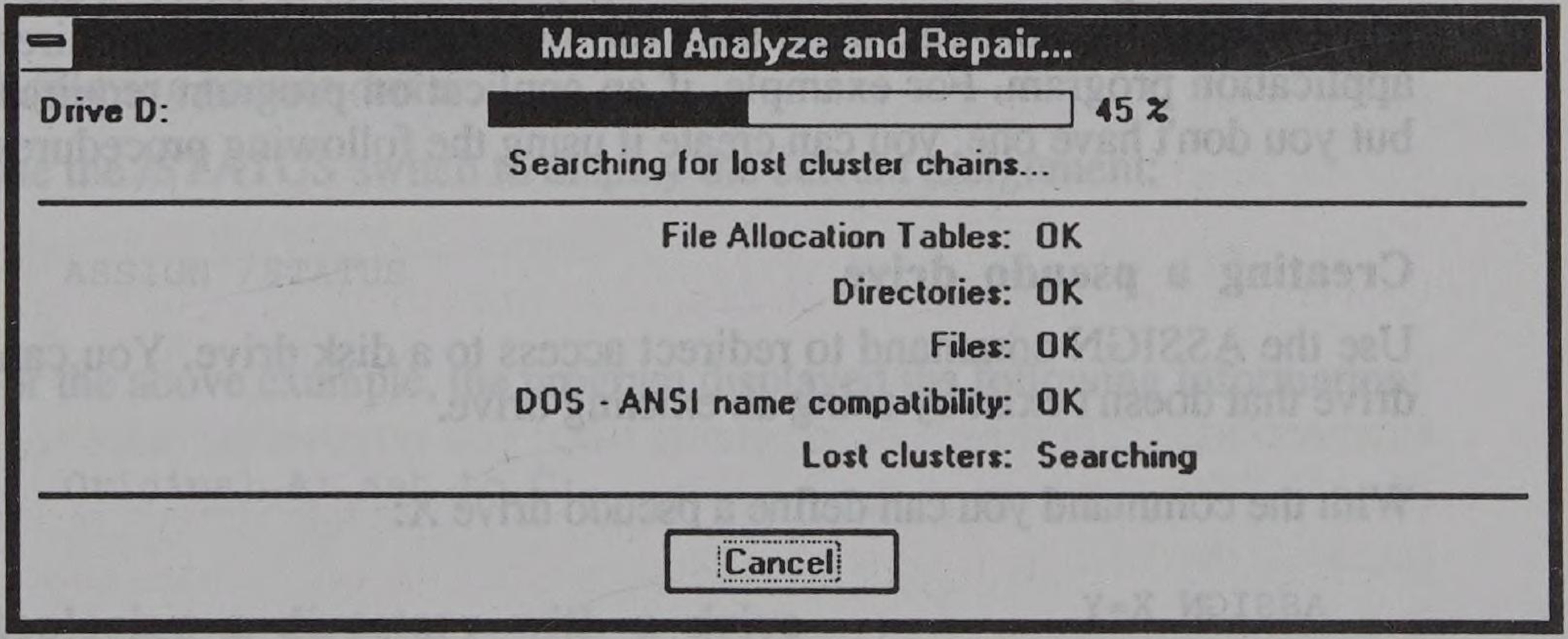
If only garbage characters appear on the screen, it's probably a program file. You can use the file manager of the MS-DOS Shell or a special utility program (such as Norton Utilities or BeckerTools) to display the contents of the file. This might help you determine what the original file was.

Lost allocation units are created when a program creates a file on the storage device, but doesn't finish the process properly because an error occurred. For example, this could occur during a power failure.

Always determine the position and kind of error (CHKDSK /V) before using /F to make corrections. This indicates which files and directories are in danger. Then you can check them with CHKDSK after repairing the error and replace them with backup copies if necessary.

Making corrections with special utilities programs

Although CHKDSK is useful for correcting errors, it may not be able to correct serious problems. If this happens you'll need to purchase a special utility program that is more powerful. The "Diskdoctor" from Norton Utilities or the "BeckerTools Recover" program from Abacus may help in salvaging damaged data.



Drive D: 45 %

Searching for lost cluster chains...

File Allocation Tables: OK
Directories: OK
Files: OK
DOS - ANSI name compatibility: OK
Lost clusters: Searching

BeckerTools Recover program finds and corrects disk errors

3.10 Redirecting Drive Access

Besides physical and logical disk drives, MS-DOS can also work with pseudo disk drives that are created by special commands. Drives and directories can also be linked. Three kinds of redirection are possible:

- Access to simulated drives; when accessing one of these pseudo drives, MS-DOS uses an existing drive instead.

```
pseudo access       real access to

E:                  A:
```

- MS-DOS simulates a subdirectory and accesses an existing drive instead.

```
pseudo access       real access to

C:\DATA             A:
```

- MS-DOS simulates a drive and accesses the subdirectory of a different drive instead.

```
pseudo access       real access to

D:                  C:\Dr_D
```

The following commands for defining pseudo drives don't work with other MS-DOS commands. Use them only for the special hardware requirements of an application program. For example, if an application program requires a drive B: but you don't have one, you can create it using the following procedures.

Creating a pseudo drive

Use the ASSIGN command to redirect access to a disk drive. You can simulate a drive that doesn't exist by using an existing drive.

With the command you can define a pseudo drive X:

```
ASSIGN X=Y
```

When you use this new drive X, MS-DOS actually accesses the existing drive Y. Substitute the desired drive letters (without colons) for X and Y. Drive Y must be a real physical drive; it cannot be a pseudo drive itself.

In order to define the pseudo drive, you may have to increase the number of possible drives in the CONFIG.SYS file using the LASTDRIVE command.

For example, if a program requires that it's run from drive A: but you would rather use the hard drive (C:), use the following command to define the assignment:

```
ASSIGN A=C
```

MS-DOS will then redirect every access of the program on drive A: to drive C:.

Limitations

You cannot use the following commands for disk drives defined by ASSIGN:

```
BACKUP
RESTORE
FORMAT
LABEL
DISKCOPY
DISKCOMP
PRINT
SUBST
```

Use ASSIGN only when absolutely necessary. Since ASSIGN is mostly used to meet special requirements of user programs, you should redirect access before calling the program and then undo your redirection when you're finished using the program.

Removing pseudo drives

Using ASSIGN without any other parameters deactivates all pseudo drives:

```
ASSIGN
```

You can display the current pseudo drives with their definitions by using the /S switch.

Displaying pseudo drives

Use the /STATUS switch to display the current assignment:

```
ASSIGN /STATUS
```

For the above example, the program displayed the following information:

```
Original A: set to C:
```

Replacing a directory with a drive

MS-DOS can treat a drive as if it were the subdirectory of a different drive. When you address this subdirectory, MS-DOS redirects access to the drive.

Use the ASSIGN command to apply MS-DOS commands to several drives at the same time by defining the drives as subdirectories on one drive.

Defining the assignment

To assign a drive to a directory, use the command:

```
JOIN drive directory
```

By doing this you can no longer access the drive with the drive letter. Instead you must access the directory specified. There are two requirements:

1. The drive cannot be the current drive when you type the command. Also, the directory specified cannot really exist on that drive.

2. The directory you specify must be empty, otherwise JOIN displays the error message "Directory not empty". If the directory still doesn't exist, JOIN automatically creates it.

The subdirectories of the drive are now available as subdirectories of the specified directory.

Example

```
JOIN A: C:\DATA
```

This command assigns the directory C:\DATA to drive A:. Now when you enter:

```
DIR C:\DATA
```

MS-DOS displays the data of the root directory from drive A:.

If the subdirectory A:\TEXTS is on drive A:, you can access it as the path C:\DATA\TEXTS.

When you use DIR to display the contents of the entire hard drive, it also displays the contents of the directory C:\DATA, which means that it displays the contents of drive A:

```
DIR C:\ /S
```

This applies to other commands that you use to edit or display files using subdirectories. There will be problems when you remove the diskette from drive A:. MS-DOS displays an error message even though you are only (apparently) accessing drive C:.

Removing the assignment

To cancel the assignment, enter the drive and /D (for **D**isjoin) as a switch following the JOIN command:

```
JOIN Disk_drive /D
```

To remove the assignment from the example above, use the following command:

```
JOIN A: /D
```

The root directory is then automatically set as the current directory. If the program created a directory for the redirection, it isn't automatically removed. Use the MS-DOS RD command to remove it.

Displaying assignment

To display all of the assignments, use the JOIN command without adding parameters:

```
JOIN
```

For our example, the display should look similar to the following:

```
A: => C:\DATA
```

Limitations

If you assign a drive to a directory from a different drive, the drive itself is no longer available. If you try to access this drive under its drive label with an MS-DOS command (such as DIR A:), you will see the error message:

```
Invalid drive specification
```

Use JOIN only when absolutely necessary and remember to undo the redirection after you're finished.

Replacing a drive with a directory

You can redirect access to a nonexistent drive to a directory. This sets up programs in a subdirectory that should be installed in a root directory. This procedure is the opposite of the JOIN command.

Defining the assignment

The instruction:

```
SUBST drive directory
```

redirects every access to the drive to the directory you specify.

In order to redirect the directory you may have to increase the number of possible drives in the CONFIG.SYS file using the LASTDRIVE command.

Example

Suppose that a particular program must be installed in the root directory but you would rather install it in the subdirectory C:\PRG of the hard drive. Use the following command line to set up a new drive D:

```
SUBST D: C:\PRG
```

Now you can enter:

```
D:
```

to switch to the new drive and install the program. Although the program thinks it is being installed in a root directory, in reality MS-DOS is placing it in the directory C:\PRG.

☞ Every time you reboot the PC you must either enter the SUBST command for drive D: or include the command line in AUTOEXEC.BAT before using the program.

Removing the assignment

To remove an assignment, specify the redirected drive with the /D switch:

```
SUBST drive /D
```

You cannot do this if the redirected drive is also the current drive in MS-DOS. Also, you shouldn't remove the assignment if there are application programs using this drive.

Displaying the assignment

Entering SUBST without any parameters causes all redirected drives and their directories to be displayed:

```
SUBST
```

For the example above, the following allocation should display:

```
D: => C:\PRG
```

Avoiding long path specifications

You can use SUBST to replace long path specifications with shorter ones. This saves entering complex path specifications. For example, if you don't have a drive D: and you frequently work with the directory C:\DATA\TEXTS\PRIVATE, you can make your work a lot easier by making this directory into a pseudo drive:

```
SUBST D: C:\DATA\TEXTS\PRIVATE
```

Now when you want to use the directory C:\DATA\TEXTS\PRIVATE, simply enter D:. The allocation may be cancelled by using the following command line:

```
SUBST D: /D
```

This only works if the directory C:\DATA\TEXTS\PRIVATE (that is D:) isn't the current directory.

Limitations

You cannot use the following commands for drives that you redirect to a directory using SUBST:

```
BACKUP
RESTORE
FORMAT
SYS
FDISK
CHKDSK
RECOVER
LABEL
DISKCOPY
DISKCOMP
```

3.11 Creating A RAM Disk

You can "borrow" part of the computer's memory (RAM) and use it as a fast RAM disk drive. To do this, you'll need a special device driver. In MS-DOS this device driver is called RAMDRIVE.SYS. If the driver isn't in the root directory, you must specify the complete path.

☞ Using PC-DOS, the driver is named VDISK.SYS.

Since a RAM disk is a lot faster than a disk drive or a hard drive, it speeds up many data intensive processes.

But every time you shut off the computer, the data stored in a RAM disk is lost. Remember to save any data stored in a RAM disk onto a diskette or hard drive before switching off the computer. Alternatively you might choose to store only non-changing data on the RAM disk so that data loss does not become a problem.

Special features of a RAM disk

A RAM disk has some advantages and some disadvantages when compared to disk drives and hard drives:

Advantages of a RAM disk

The most important advantage of a RAM disk is the speed with which the computer can access the data. The speed is greater than even the fastest hard drives.

- You can load programs from the RAM disk very quickly. If you use a program frequently, simply copy it to the RAM disk and load it from there.

  It's best if you extend the search path by the drive specification of the RAM disk. First, include this in the search path so that you can load programs from the RAM disk.

  For example, if D: is the drive letter of your RAM disk and MS-DOS is in the directory C:\DOS of the hard drive, the search path in the AUTOEXEC.BAT might look similar to the following:

  ```
  PATH D:\;C:\DOS
  ```

- You can retrieve data from a RAM disk faster than from a diskette or hard drive. If you use the same files often, copy them to the RAM disk and always load them from there.

Since the contents of the RAM disk are lost after you switch off the computer or press reset, you shouldn't edit or make changes to data on the RAM disk without saving a copy of the data to a permanent storage device.

- Many programs use temporary files, in which information is stored for a short time. These programs often place the temporary files in a special directory if you define an environment variable (TMP or TEMP) which contains the desired directory.

  Setting up a RAM disk and defining it as a directory for temporary files can increase the speed of your programs significantly.

  For example, in order for WORD 5.0 or 5.5 to place all of its temporary files on RAM disk D:, type the following command line before calling WORD 5.0:

  ```
  SET TMP=D:\
  ```

  It's best to include this line, which sets the TMP environment variable, in the AUTOEXEC.BAT file.

- If your PC doesn't have a hard drive, you can copy the command interpreter and the most important external commands to a RAM disk. By doing this you won't have to insert an MS-DOS diskette several times. You should do this automatically in the AUTOEXEC.BAT file.

  You should also include the special system variable COMSPEC for the new command interpreter in the RAM disk so that MS-DOS knows where it can load the command interpreter. For example, if drive D: is the RAM disk, define the environment variable COMSPEC with the following command line:

  ```
  SET COMSPEC=D:\COMMAND.COM
  ```

Disadvantages of a RAM disk

The biggest disadvantage of a RAM disk is that the data is stored in the memory of the computer (RAM):

- Switching off the computer, using the reset button, performing a warm boot with Ctrl + Alt + Del, power failures and serious program errors all cause the data in the RAM disk to be lost forever. So it's very dangerous to edit and make changes to data in the RAM disk. It's safer to store data to the RAM disk which is also stored on a diskette or hard drive.

- Since the RAM disk is completely empty every time you reboot, first you must copy data or programs to the RAM disk. That's why you should place data, that you use often, on the RAM disk.
- If your computer doesn't have extended or expanded memory, the memory for the RAM disk is reduced to the available main memory. A RAM disk with 256K of memory (a lot less than the capacity of a 360K diskette) won't provide enough storage for extensive application programs.

  Reserve a maximum of 128K of memory for the RAM disk on PCs that don't have extended or expanded memory. Even with this you can still have problems. If your computer has a hard drive, but not extended or expanded memory, use the hard drive instead of a RAM disk.

Alternatives to a RAM disk

Cache programs are an alternative to the RAM disk. They also use part of the main memory for storing data but don't function like a disk drive.

A cache program consists of taking information, that's already read, from the disk drive and temporarily storing it in the memory. Then if the information is needed again, it is taken from RAM, which is much faster than the hard drive. When the reserved amount of memory for the cache is filled and more information needs to be stored, available space must be made in the cache.

Using a cache has two distinct advantages over using a RAM disk:

- Unlike a RAM disk, you don't have to copy the data to memory before it's used. The cache program does this automatically the first time it retrieves the data from the hard drive.
- The cache program determines which data will be stored temporarily in the main memory. You must make this decision when using a RAM disk. For example, it isn't very useful to place the MS-DOS FORMAT command in the RAM disk when you use the DISKCOPY command more often. However, the cache program would notice this and automatically store the DISKCOPY command in RAM.

Creating a RAM disk with standard default

Insert the following line in the CONFIG.SYS of your system diskette or hard drive:

```
DEVICE=C:\DOS\RAMDRIVE.SYS
```

☞ Remember to include the complete search path for your system.

The next time you start your PC you'll have a new drive, which has a capacity of 64K and can contain 64 files or directories. When you start your PC the screen displays this information.

Creating a RAM disk with special defaults

When you define the RAM disk, you can set the capacity, the size of the sector and the number of possible entries. To do this use additional parameters when defining the RAM disk in the CONFIG.SYS:

```
DEVICE=C:\DOS\RAMDRIVE.SYS disk_size sector_size entries
```

Disk_size	Specifies the size of the RAM disk in K. The default size is 64K; 16K is the minimum size.
Sector_size	Here you specify the least amount of space a file occupies. Larger files always take up a multiple of this sector size. Possible values are 128, 256, 512 and 1024. The RAM disk has a default sector size of 512 bytes.
Entries	Determines the number of files and/or directories that you can put on the RAM disk (4 to 1024). The default value is 64. RAMDRIVE.SYS automatically adapts this value to the sector size.

Placing a RAM disk in extended memory

A RAM disk can use extended memory. This enlarges the available main memory considerably.

RAM disk in Extended Memory

To use Extended Memory for storage, specify /E as your additional switch. The line in the CONFIG.SYS is as follows:

```
DEVICE=C:\DOS\RAMDRIVE.SYS disk_size sector_size entry /E
```

To use the 384K extended memory available on an AT with 1 Meg for the RAM disk and to have sector size and file entries work with default values, use the following line in the CONFIG.SYS:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 384 /E
```

Remember to install the software for memory extension before the RAM disk. Since you ordinarily do this by calling the appropriate driver program in the CONFIG.SYS, call the driver for the memory before defining the RAM disk.

Define the RAM disk in the CONFIG.SYS file before other driver programs. The drivers EMM386.SYS and HIMEM.SYS are exceptions to this rule. EMM386.SYS allows you to use the extended memory in a 386 as Expanded

Memory. HIMEM.SYS allows you to use extended memory in ATs as XMS-Memory and makes High Memory (special memory area) available. Both of these drivers must be installed before the RAM disk.

☞ The new RAM disk driver RAMDRIVE.SYS, which is included with MS-DOS 5.0, follows the rules for Extended Memory (XMS-specification) so that you can only use it if you install HIMEM.SYS beforehand.

As soon as you install HIMEM.SYS, the program automatically reserves 64K of memory for High Memory. This area is no longer available to the RAM disk. On a computer with 1 Meg memory (384K Extended Memory) you would only have 324K available for the RAM disk.

RAM disk in Expanded Memory

You can also use expanded memory that conforms to the LIM 4.0 specification for the RAM disk. Use /A as your switch.

To use 1 Meg (1024K) of existing expanded memory for the RAM disk and work with default values for the sector size and the file entries, type the following line in the CONFIG.SYS:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 1024 /A
```

☞ You cannot place a RAM disk in the Expanded Memory of the VDISK.SYS (PC-DOS).

Removing the RAM disk

To remove a RAM disk, delete the command line from CONFIG.SYS and reboot the computer. A RAM disk can't be removed without rebooting the computer.

3.12 Speeding Up Your Hard Drive Access

Although hard drives are faster than disk drives, extensive loading and memory processes still require a considerable amount of time. Although there are extremely fast hard drives, these drives are very expensive and not available in low priced computers.

The basics of speed up

There are ways to speed up access on a relatively slow hard drive. One way is to use the hard drive as little as possible. This may sound strange, but it actually is a powerful method for accelerating the hard drive. In order to do this, you can use a cache.

A cache is an area of memory in which copies of recently used data of a disk are stored. Since the data is kept in memory, you don't have to access the disk drive as often. By cutting down accesses to the disk drive, you can speed up the performance of an application significantly.

Since memory is limited, it isn't possible to store all of the data of a disk. So as cache memory fills up, copies of the least recently referenced data are replaced by more recently referenced data.

A cache program controls and organizes this technique for speeding up performance.

Speeding up using BUFFERS

The CONFIG.SYS file may contain a statement BUFFERS=XX, where XX specifies the number of 512 byte buffers that MS-DOS will use to temporarily store data. If this statement is omitted, DOS chooses a default value that depends on several other factors. Since the default value is usually small, this may cause degraded performance.

Therefore, we recommend that you insert a statement, such as BUFFERS=30 for a computer with a 30 Meg hard drive. Adding this statement to the CONFIG.SYS file will take effect the next time you boot your computer. When you restart, you will immediately notice that your hard drive operates at a higher speed.

BUFFERS has some limitations, For one, a value higher than 99 can't be entered. This results in a maximum of 50K of cache memory. Also, the method used for replacing older information with information just being read isn't very effective. Special cache programs can improve the effect considerably.

Speeding up using cache programs

The cache program SMARTDRV.SYS also is part of MS-DOS 5.0. Only extended or expanded memory can be used with SMARTDRV.

- Extended and Expanded Memory can be used for storing data. An AT with 1 Meg memory can use 384K as cache memory. The SMARTDRV auxiliary program itself only requires 14604 bytes in main memory. Available main memory is only slightly reduced.

- SMARTDRV replaces older cache contents with newer contents according to a convenient system. The data you need will most likely be in the buffer.

The more memory a cache program has, the more effective it can be. If your PC doesn't have extended or expanded memory, the memory a cache program uses would be taken from main memory. In a short time, you wouldn't be able to load any large application programs. Even a smaller size cache can lose some of its speed advantages when memory intensive application programs frequently move information from the main memory to the hard drive. This wouldn't be necessary if more memory was available. We recommend using SMARTDRV only if your computer has extended or expanded memory.

Using Cache in Extended Memory

Since SMARTDRV is a device driver, start it from the CONFIG.SYS file. To use Extended Memory for temporarily storing data and programs, simply specify the size of the memory in kilobytes. On an AT with 1 Meg of memory, use 384K of Extended Memory by typing the following command line in the CONFIG.SYS file:

```
DEVICE=C:\DOS\SMARTDRV.SYS 384
```

☞ If you use the Extended Memory Manager HIMEM.SYS, you cannot define SMARTDRV until you define HIMEM.SYS. HIMEM.SYS reserves the first 64K of extended memory as High Memory, leaving that much less memory available for Extended Memory.

Using Cache in Expanded Memory

To use Expanded Memory, use the /A switch. If you want to use 256K of Expanded Memory for temporary storage, type the following command line:

```
DEVICE=C:\DOS\SMARTDRV.SYS 256 /A
```

☞ The SMARTDRV statement in the CONFIG.SYS file must follow the Expanded Memory driver statement.

If you use the EMM386.SYS device driver on an 80386 system to emulate Expanded Memory, which converts existing Extended Memory to Expanded Memory, don't assign this memory to SMARTDRV. This unnecessarily reduces the access speed to this memory. Reduce the Expanded Memory by the value that you want to use for temporary storage and assign this memory as Extended Memory.

Example

You have 1 Meg of Extended Memory and have been using the entire extended memory as Expanded Memory. Use the following statement in the CONFIG.SYS file:

```
DEVICE=C:\DOS\EMM386.SYS 1024
```

Now you want to use 512K with SMARTDRV for temporary storage. Reduce the Expanded Memory to 512K and use only 512 as a parameter when you use SMARTDRV in the CONFIG.SYS file:

```
DEVICE=C:\DOS\EMM386.SYS 512
DEVICE=C:\DOS\SMARTDRV.SYS 512
```

By doing this, you'll be using 512K for temporary storage and still have 512K of Expanded Memory for application programs.

Acceleration using FASTOPEN

Using FASTOPEN speeds up repeated access to files and directories. Ordinarily MS-DOS searches the directory structure of the hard drive every time you need a directory or file.

FASTOPEN remembers the locations of files and directories on the storage device so that the access is much faster for subsequent requests. This is especially useful for working with application programs, such as database programs and word processors, that open and close files frequently.

To speed up repeated access to files, after FASTOPEN enter the drive and desired number of files:

```
FASTOPEN X:=nnn
```

X represents an existing hard drive and nnn represents the number of files. To have the program remember up to 50 file positions for the hard drive (C:), use the following command:

```
FASTOPEN C:=50
```

You should also insert the line in the AUTOEXEC.BAT file so that FASTOPEN is available immediately after you start the computer.

FASTOPEN requires about 48 bytes for each file it remembers. So in the example line above, only 2,400 bytes of main memory are used. If your computer has Expanded Memory, FASTOPEN can also store the file positions in the Expanded Memory. Type the /X switch:

```
FASTOPEN C:=50 /X
```

Instead of being stored on a hard drive contiguously, a file will often consist of several consecutive areas. Information on how the individual elements of the file go together is stored in the File Allocation Table (FAT). However, MS-DOS needs time to determine the arrangement every time you access a file.

In order for FASTOPEN to remember this arrangement, add the following command:

```
FASTOPEN X:=(nnn,mmm)
```

In the command line, mmm represents the number of allocations; you can use values between 1 and 999. Each entry requires 16 bytes of memory.

Example

If you want FASTOPEN to remember 50 file positions and 25 allocations on drive C:, use the following line:

```
FASTOPEN C:=(50,25)
```

You can also specify the /X switch if your computer has Expanded Memory. This will take some of the responsibility off the main memory.

3.13 MS-DOS Internals

If you feel overwhelmed by the title of this section, feel free to skip it. Here we'll present information for users that just need to know more.

Basically, MS-DOS manages storage devices using block driver programs that can read or write a block of data at a time. This is why MS-DOS also arranges storage devices into blocks (sectors or clusters).

A storage device is set up in the following order:

```
Boot sector
FAT
Root directory
Actual data
```

Structure of storage devices

Diskettes are simply flexible plates (disks) that are coated with a magnetic material. When MS-DOS prepares a diskette (formats) it divides the diskette into tracks and sectors.

Tracks and sectors

Let's use a 360K, 5 1/4 inch diskette as an example. Computers can't simply write on the diskette. If they did this, they would have trouble finding and recognizing information that belongs together. So a diskette is divided into tracks. The computer numbers the tracks consecutively, which results in a specific arrangement.

However, even this arrangement isn't flexible. By itself this allows you to write only 40 different groups of data in the 40 tracks.

So tracks are further divided into blocks (9 blocks on a 360K diskette) called sectors. Using this arrangement and a few additional pieces of information, MS-DOS can store 512 characters in one of these blocks. This would appear to be the minimum size for storing information as a unit. Since there are 40 tracks with 9 sectors on each side of the diskette, there are a total of 720 sectors numbered from 0 to 719.

But MS-DOS, instead uses two sectors. This unit of allocation is called a cluster. So for a 360K diskette one cluster refers to two sectors or 1024 characters. The reason for this is that when MS-DOS was introduced, diskettes were only single sided (180K) and contained 360 sectors. So the management information on the diskette was only set up for 360 elements.

When the capacity was doubled to 360K, only the old space was reserved for the management information in order to remain compatible with older versions of

DOS. Two sectors were integrated into one unit, called a cluster. A file always takes up at least one cluster. Even a file containing a single character uses 1024 bytes on a 360K diskette. This problem wasn't resolved until 1.2 Meg diskettes were developed. MS-DOS reserves more room for managing information on 1.2 Meg diskettes, which have clusters composes of a single sector with 512 bytes.

Division into boot block, FAT and directory

Besides the data, a diskette contains a couple of special areas for managing this stored information.

Table of contents directory

MS-DOS always stores data on a diskette as a file. You can then refer to this data by its filename. In order to do this, MS-DOS must save information about each file. MS-DOS does this in the disk directory. The disk directory is a collection of sectors reserved for managing files.

Basically, for each file in the directory, MS-DOS stores the name, date and time it was created, the size of the file and the location of the first sector of data on the diskette.

The directory of a 360K diskette starts at the fifth sector. A small section of this directory would look like the following:

```
No    contents as hexadecimal numbers            character representation
-------------------------------------------------------------------------
0000  49 4F 20 20 20 20 20 20 53 59 53 27 00 00 00 00  IO            SYS
0016  00 00 00 00 00 00 00 18 81 15 02 00 BE 82 00 00
0032  4D 53 44 4F 53 20 20 20 53 59 53 27 00 00 00 00  MSDOS     SYS
0048  00 00 00 00 00 00 00 18 81 15 23 00 10 92 00 00
0064  43 4F 4D 4D 41 4E 44 20 43 4F 4D 20 00 00 00 00  COMMAND COM
0080  00 00 00 00 00 00 00 18 C7 15 48 00 4B A0 00 00
```

Notice that each entry for a file contains 32 bytes, which requires two printed lines. The first 11 bytes contain the filename (8 bytes) and the extension (3 bytes), without the period between the name and the extension. This results in the following structure for a file entry:

Byte	Length	Description
00-07	8	filename, padded with spaces
08-10	3	extension, padded with spaces
11	1	attribute, special features of a file
12-21	10	reserved
22-23	2	time of creation or last change
24-25	2	date of creation or last change
26-27	2	location of first cluster of file data
28-31	4	size of the file

When the computer stores numbers, the lower value digits are on the left while the higher value digits are on the right (exactly the opposite of our decimal system). The last four bytes of the file size under IO.SYS "BE 82 00 00" correspond to "00 00 82 BE" in hexadecimal notation. Converted to our decimal system they read 33470. Attributes, date and time are even more complicated in their structure.

If you want to examine the directory of a 360K diskette, use a utility program such as PC Tools or Norton Utilities, or use the MS-DOS DEBUG command. Insert the 360K diskette into drive A: and make drive A: the current drive. After starting DEBUG type the following command lines:

```
L 0100 0 5 1
D 0100
```

This causes DEBUG to load the fifth sector into the memory and display it on the screen. Press Q and Enter to leave DEBUG.

FAT

In addition to a directory, two additional pieces of information are required to manage the data of a diskette: which sectors belong to a file and which sectors are still unused. This information is maintained in the File Allocation Table (FAT), which is where the storage units of a diskette are assigned to a file.

The FAT of a 360K diskette begins at sector 1, which is the second sector (start counting at 0), and is two sectors (1024 bytes) long. A section from the beginning of the FAT would look like the following:

```
0000  FD FF FF 03 40 00 05 60 00 07 80 00 09 A0 00 0B
0016  C0 00 0D E0 00 0F 00 01 11 20 01 13 40 01 15 60
0032  01 17 F0 FF 19 A0 01 1B C0 01 1D E0 01 1F 00 02
0048  21 20 02 23 40 02 25 60 02 27 80 02 29 A0 02 2B
0064  C0 02 2D E0 02 2F 00 03 31 20 03 33 40 03 35 F0
0080  FF 37 80 03 39 A0 03 3B C0 03 3D E0 03 3F 00 04
0096  41 20 04 43 40 04 45 60 04 47 80 04 49 A0 04 4B
0112  C0 04 4D E0 04 4F F0 FF 00 00 00 00 00 00 00 00
```

We didn't list the characters that would be displayed on the right side of the screen. These are nonsense characters that have no real significance. The filenames are not stored there.

To display the beginning of the FAT of a 360K diskette, use the MS-DOS DEBUG command with the following command line:

```
L 0100 0 1 1
D 0100
```

It's difficult to interpret the FAT. On diskettes, not one or two bytes, but 1.5 bytes contain information (12 bits, 1 byte = 8 bits). The first line of our sample display consists of the following entries:

```
0000  FDF FFF 034 000 056 000 078 000 09A 000 0BC
```

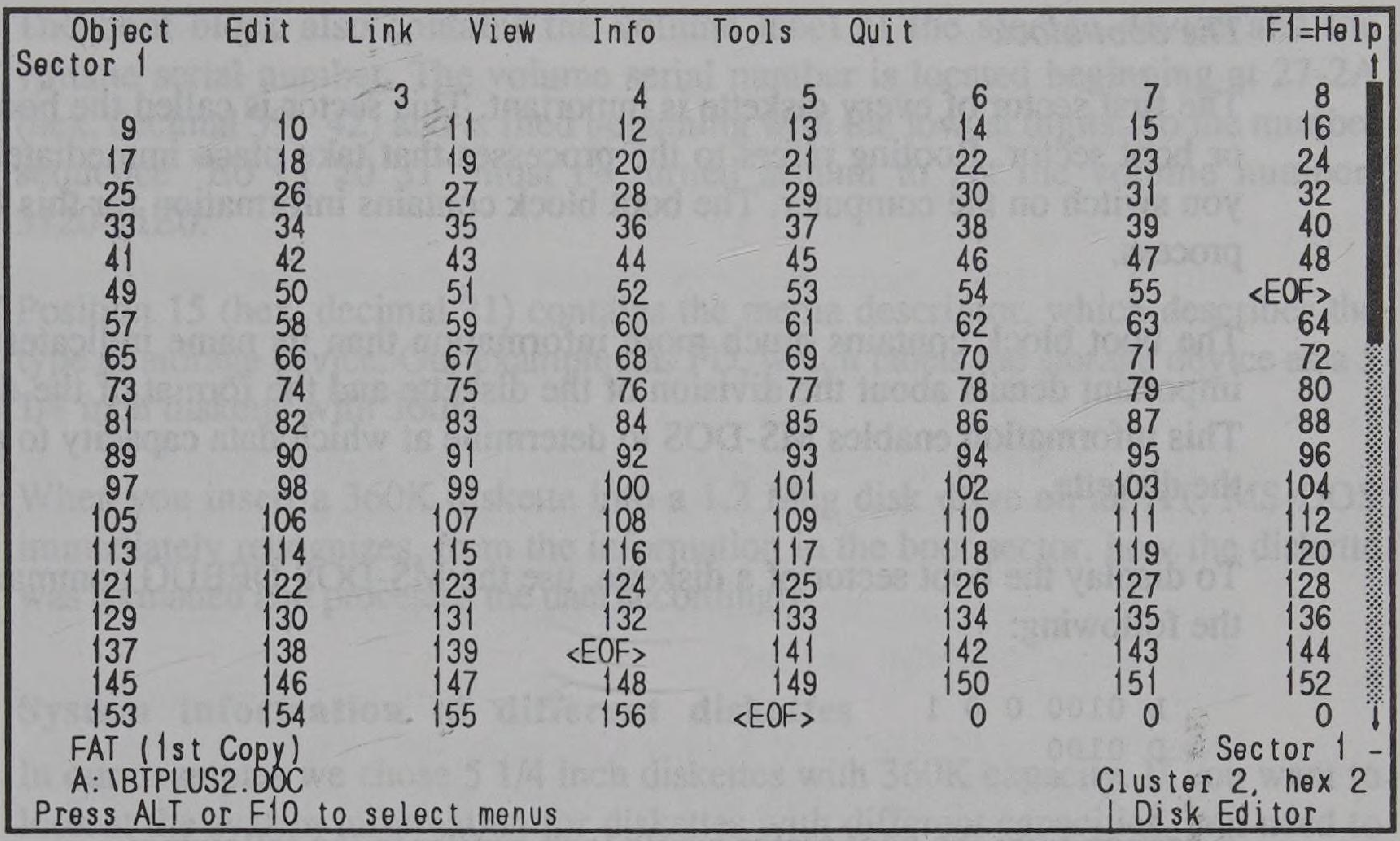
The first entry (No. 0), with the hexadecimal value FDF, labels the diskette as a 5 1/4 inch diskette with two sides, 40 tracks per side and 9 sectors per track (360K). An AT diskette with 80 tracks per side and 15 sectors per track (1.2 Meg) would have the hexadecimal value of F9F. This information is also called the media descriptor. A value describes the medium being used. The second entry (No. 1) is reserved and always contains the value FFF.

The subsequent entries label the clusters where the data is filed. Let's return to the structure of a file entry in the table of contents directory. Bytes 26 to 27 specified the first cluster of the file. For the IO.SYS file entry, both bytes had "02 00" (the value 2). The file data begins with cluster number 2 (sector no. 12, the first sector after the table of contents directory).

Cluster 2 always contains the number of the cluster where the file continues. You cannot use this number as a 12 bit value (1.5 bytes) because MS-DOS files the values differently. In our example, the value seems to be 34 (hex) but it's actually 003 (the third cluster). The previous line should look as follows:

```
0000  FDF FFF 003 004 005 006 007 008 009 00A 00B
```

We're not really interested in where the data is located, since MS-DOS looks for the clusters that we use when we edit a file. You can use special utility programs, such as Norton Utilities, so that you don't have to convert the 12 bit values. Norton Utilities has a special FAT Editor. So whenever you try to display a sector of the FAT, the FAT Editor automatically shows the numbers correctly. <EOF> marks the end of a file.



```
  Object  Edit   Link    View    Info    Tools    Quit                F1=Help
 Sector 1
                           3        4        5        6        7        8
      9       10       11       12       13       14       15       16
     17       18       19       20       21       22       23       24
     25       26       27       28       29       30       31       32
     33       34       35       36       37       38       39       40
     41       42       43       44       45       46       47       48
     49       50       51       52       53       54       55    <EOF>
     57       58       59       60       61       62       63       64
     65       66       67       68       69       70       71       72
     73       74       75       76       77       78       79       80
     81       82       83       84       85       86       87       88
     89       90       91       92       93       94       95       96
     97       98       99      100      101      102      103      104
    105      106      107      108      109      110      111      112
    113      114      115      116      117      118      119      120
    121      122      123      124      125      126      127      128
    129      130      131      132      133      134      135      136
    137      138      139    <EOF>      141      142      143      144
    145      146      147      148      149      150      151      152
    153      154      155      156    <EOF>        0        0        0
  FAT (1st Copy)                                              Sector 1 -
 A:\BTPLUS2.DOC                                           Cluster 2, hex 2
 Press ALT or F10 to select menus                          |Disk Editor
```

FAT Editor of Norton Utilities

Instead of providing the next data portion of a file, some cluster values describe different special conditions:

000	Cluster is free
FF0 - FF6	Reserved cluster
FF7	Cluster is defective and cannot be used
FF8-FFF	Last cluster of a file <EOF>

The example we've been using is for a 360K diskette which automatically has a FAT with 12 bit values (1.5 bytes or 12 bit FAT). On hard drives, the FAT usually contains 16 bit entries and a cluster may be composed of 4 or even 8 sectors.

The directory of a diskette has a file entry for each file with the number of the first cluster of the file. This allows MS-DOS to read the first two sectors (on a 360K diskette) directly. To find other data on the diskette, MS-DOS follows the "linkage" within the FAT until a cluster number between FF8 and FFF signifies the end of the file.

It's obvious that the directory and the FAT play a significant role in working with files. If there are problems, you can no longer work with the data of the diskette. Because of this, MS-DOS even places a copy of the FAT on the diskette. This copy follows the first FAT in sectors 3 and 4.

The boot block

The first sector of every diskette is important. This sector is called the boot block or boot sector. Booting refers to the processes that take place immediately after you switch on the computer. The boot block contains information for this booting process.

The boot block contains much more information than its name indicates. It has important details about the division of the diskette and the format of the diskette. This information enables MS-DOS to determine at which data capacity to prepare the diskette.

To display the boot sector of a diskette, use the MS-DOS DEBUG command with the following:

```
L 0100 0 0 1
D 0100
```

A section from the boot sector would look similar to the following:

```
0000  EB 3C 90 4D 53 44 4F 53 35 2E 30 00 02 02 01 00    MSDOS5.0
0016  02 70 00 D0 02 FD 02 00 09 00 02 00 00 00 00 00
0032  00 00 00 00 00 00 29 E0 11 20 31 41 52 42 45 49            WORK
0048  54 44 49 53 4B 20 46 41 54 31 32 20 20 20 FA 33  DISK FAT12    •
0064  C0 8E D0 BC 00 7C 16 07 BB 78 00 36 C5 37 1E 56
0080  16 53 BF 3E 7C B9 0B 00 FC F3 A4 06 1F C6 45 FE
0096  0F 8B 0E 18 7C 88 4D F9 89 47 02 C7 07 3E 7C FB
0112  CD 13 72 79 33 C0 39 06 13 7C 74 08 8B 0E 13 7C
0128  89 0E 20 7C A0 10 7C F7 26 16 7C 03 06 1C 7C 13
```

Beginning at position 3 (starting at 0), you can clearly recognize the name and version number of MS-DOS 5.0. The other areas of the boot sector usually contain the following text:

```
Non-System diskette or disk error
Replace and press any key when ready
IO      SYSMSDOS   SYS
```

It's interesting that there is a command at the beginning of the boot sector (in the first three bytes) to run a program that starts at a later position. In our example, this program begins at position 013E (hex). It loads both of the hidden system files (in our example IO.SYS and MSDOS.SYS) and in so doing, starts MS-DOS 5.0. The names of both files are in part of the boot sector.

There is important information about the storage device in the boot sector from position 0B to 1D (hex). For example, position 0B (decimal 11) provides the number of bytes per sector. In our example it is "00 02" or 0200 (hex), which means 512 bytes per sector.

The boot block also contains the volume label of the storage device and the volume serial number. The volume serial number is located beginning at 27-2A (hex, decimal 39 - 42) and is filed beginning with the lowest digits. So the number sequence "E0 11 20 31" must be turned around to get the volume number: 3120-11E0.

Position 15 (hex, decimal 21) contains the media descriptor, which describes the type of storage device. Our example has FD, which labels the storage device as a 5 1/4 inch diskette with 360K.

When you insert a 360K diskette into a 1.2 Meg disk drive on an AT, MS-DOS immediately recognizes, from the information in the boot sector, how the diskette was formatted and processes the data accordingly.

System information of different diskettes

In our examples we chose 5 1/4 inch diskettes with 360K capacity. If you want to look at the system information for diskettes with different capacities, you need to know the right sectors:

	360K	1.2 Meg	720K	1.44 Meg
Boot sector	0	0	0	0
FAT	1	1	1	1
Root directory	5	15	7	19

After calling DEBUG, load a sector into the memory with the following command line:

```
L address disk_drive start_sector number
```

For example, if you specify 0100 for the address, you can look at the data by typing:

```
D 0100
```

For disk drives, specify 0 for drive A: and 1 for drive B:. For your starting sector specify the desired sector on the diskette, using the values from the table above. For number you can also enter a value larger than 1; then the program loads several sectors into the memory, one after the other.

☞ To exit DEBUG simply type Q at the prompt.

What happens when you format a diskette

The division of the diskette into tracks and sectors and the management information (directory, FAT and boot block) are created during FORMAT. That's why you can't use a new diskette until it has been formatted.

During this formatting MS-DOS checks the diskette for defective areas and excludes them from data storage. (The clusters belonging to these defective areas are labelled FF7: Cluster is defective and excluded from use.)

This is why there is a difference between the amount of data that will fit on a diskette and the total capacity. The screen automatically displays this information after the formatting process or you can display this information by using the CHKDSK command. If the diskette contains defective regions, they appear after the message "defective_sectors".

When you use FORMAT to format a diskette, the command writes new, empty sectors to the diskette. That's why any information previously stored is lost when you format.

Differences in various disk drives

Since MS-DOS uses different kinds of management information, when formatting diskettes you must specify the various disk drive types and capacities.

360K disk drive (5 1/4 inch)

When you format a 360K diskette, it is divided into 40 tracks on each side. Each track is subdivided into 9 blocks (sectors). MS-DOS can file 512 bytes (characters) in one of these sectors. Multiplying 40 tracks * 9 sectors * 512 characters *2 sides results in 368,640 characters. Since one K contains 1,024 bytes, you have 360K storage capacity.

This diskette contains 720 sectors and 360 clusters. After subtracting the management information (boot sector, FAT and directory), you still have 354 clusters available for data storage. So a 360K diskette can store a maximum of 354 files.

The root directory is 7 sectors long and since each sector can contain 16 file entries with 32 bytes, there is a maximum of 112 files in the root directory. Because they are placed in the root directory as special file entries, the subdirectories and the name of the storage device are included.

1.2 Meg disk drive (5 1/4 inch)

1.2 Meg disk drives can store more than three times as much as a 360K diskette. Each side of the diskette is divided into 80 tracks and 15 sectors. You need special diskettes, called DSHD (double sided, high density).

A 1.2 Meg diskette contains a total of 2,400 sectors and 2,400 clusters. It has 2,400 clusters because each cluster has 512 bytes due to the expanded FAT (7 sectors instead of 2). After subtracting the management information (boot sector, FAT and directory), there are still 2,371 clusters available for data storage. A 1.2 Meg diskette has a root directory with 14 sectors (twice as many as a 360K diskette) and has room for 224 files in the root directory.

When reading and writing data, the disk drive can determine what kind of a diskette has been inserted by checking the media descriptor in the boot sector of the diskette. This media descriptor contains exact information about the capacity of the diskette. When formatting a diskette, a 1.2 Meg disk drive always formats diskettes with the highest possible capacity (1.2 Meg). However, you can use special parameters with the FORMAT command to change this.

720K disk drive (3 1/2 inch)

The 720K capacity of a 3 1/2 inch diskette with double density is a result of having twice as many tracks written as a 360K diskette (80 instead of 40). It has the same number of sectors (9) as a 360K diskette. A 720K diskette contains 3 sectors per FAT and 2 sectors per cluster. Out of 720 clusters, you can use 713 clusters for data storage. Since the root directory contains 7 sectors, you can have a maximum of 112 entries (files or subdirectories) in the root directory.

1.44 Meg disk drive (3 1/2 inch)

The higher storage capacity of a 1.44 Meg disk drive results from having 18 sectors per track. If we multiply 80 tracks * 18 sectors * 512 bytes * 2 sides, the result is 1,474,560 bytes = 1.44 Meg. Similar to 1.2 Meg diskettes, a cluster contains one sector with 512 bytes. There are a total of 2,880 clusters with 2,847 clusters available for data storage; MS-DOS requires the rest for management. The root directory consists of 14 sectors and can contain 14 * 16 =224 files.

These disk drives are similar to 1.2 Meg disk drives. You can operate them with high capacity and you can also edit 720K diskettes. By the media descriptor in the boot block, MS-DOS can tell which capacity the diskette was formatted for and treats it accordingly.

Formatting diskettes as system diskettes

When you add the /S switch to the MS-DOS FORMAT command, MS-DOS automatically copies the operating system to the new diskette. The operating system consists of two hidden files and the command interpreter COMMAND.COM. In MS-DOS these hidden files are called IO.SYS and MSDOS.SYS and in PC-DOS by IBM they are called IBMBIO.COM and IBMDOS.COM.

You can start your PC using this system diskette. To boot your computer from a system diskette, place the diskette in drive A: and close the drive lever.

Remember that simply copying the hidden files to a diskette won't make it a system diskette. There are two crucial requirements:

- Both the IO.SYS and MSDOS.SYS files must be the first two files in the root directory of the diskette.
- Both files must be located in the first clusters of the storage device.

Leaving room for the system

If the /B switch is used when formatting a diskette, MS-DOS reserves room for the system. Then you can copy the system to the diskette with the MS-DOS SYS command. In this procedure, the FORMAT command creates two files with the correct names and sizes. This reserves, on the diskette, the amount of space necessary for the system. MS-DOS 5.0 creates an IO.SYS file with approximately 35,000 bytes and an MSDOS.SYS file with 37,000 bytes. Then the SYS command replaces these files with files of the same name containing the correct contents.

Diskette volume label and volume serial number

After formatting a diskette, you can specify a volume label for the storage device. You can also do this later with the LABEL command. The volume label of a storage device is stored in almost the same way as a filename. The root directory of a diskette looks like the following:

```
0000  49 4F 20 20 20 20 20 20 53 59 53 26 00 00 00 00  IO        SYS
0016  00 00 00 00 00 00 64 56 35 15 02 00 B8 88 00 00
0032  4D 53 44 4F 53 20 20 20 53 59 53 26 00 00 00 00  MSDOS     SYS
0048  00 00 00 00 00 00 65 56 35 15 47 00 88 90 00 00
0064  41 52 42 45 49 54 53 44 49 53 4B 28 00 00 00 00  WORKDISK
0080  00 00 00 00 00 00 81 58 35 15 00 00 00 00 00 00
```

You can see that the label of the diskette (in our example WORKDISK) is filed similar to the two system files, except that the filename and its extension aren't separated.

MS-DOS must be able to distinguish a filename from the diskette name. It can distinguish the two by using the attribute, beginning at position 11, in the file entry. For the first two file entries the value is 26 (hex). The diskette name has an attribute of 28. The attribute byte consists of 8 bits, 6 of which are used to differentiate different file entries:

Bit 0 = 1 File is write protected.

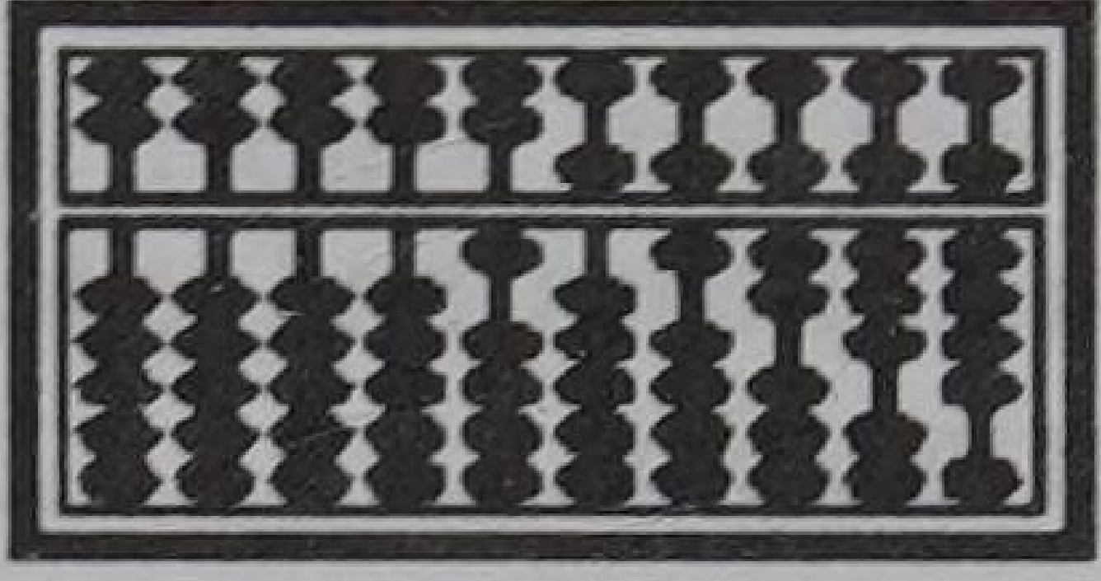
Bit 1 = 1	File is hidden, not usually displayed.
Bit 2 = 1	System file.
Bit 3 = 1	This entry is the name of the storage device.
Bit 4 = 1	This entry is not a file, but a subdirectory.
Bit 5 = 1	This file has been changed since the last backup.

If you break both attributes down according to this system, you get the following result:

Hex	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
26	1	0	0	1	1	0
28	1	0	1	0	0	0

The first two files have been changed since the last backup. They are hidden and labelled as system files. The third entry has also been changed and it's marked as a storage device name.

Chapter



4

Order Through Directories

In this chapter we'll present a general overview of hard drives and diskettes.

Every hard drive or diskette has at least one directory, called the root directory. By convention, the root directory is identified by the backslash symbol (\). The directory currently in use is indicated, after the drive identifier and the backslash symbol.

The system prompt is displayed after you start your computer. In the following example, the current directory is in the root directory of Drive C:.

```
C:\>
```

If you insert a diskette in disk drive A: and change the current drive to drive A:, the system prompt displays:

```
A:\>
```

☞ A colon, backslash and the greater than symbol are displayed after the drive letter because MS-DOS presets this in the AUTOEXEC.BAT using the PROMPT PG statement during installation. This can be changed to suit your needs.

Why use directories?

Directories are mainly used when working with hard drives. If only the root directory exists, then MS-DOS always stores all files in the root directory. Since hard drives usually have a very high storage capacity, an unwieldy number of files might accumulate in the root directory. Obviously this makes it difficult to keep track of files.

To avoid confusion in your root directory, MS-DOS lets you define and work with subdirectories, which help you better manage the files.

Directories can be compared to drawers in a desk. One drawer can be used to store personal letters and another could store business letters. The drawer containing the business letters could also be subdivided into a section containing letters concerning financial matters and a section containing letters regarding your legal matters.

Directories on diskettes, and especially on your hard drive, serve the same purpose as the drawers in a desk.

What are directories?

Directories are "logical", which means that they are manageable parts of a storage device. You can structure or organize a storage device by arranging or placing a collection of files in a named "drawer" or directory. The named "drawer" is similar to the label on an office file cabinet - it describes what is inside the drawer.

You will probably choose to store files that belong together in a particular directory on your hard drive. For example, you could put all of your personal letters in a directory called PERSONAL. By doing this, you can easily locate and backup files that belong together. Also, you might choose to place files that belong to a certain application program in one directory.

Since the application program will often require different files, placing all of the necessary files for a particular application program in one directory guarantees that the program will be able to easily find them. Often when you install these programs, you will be prompted to place the files belonging to the program in a particular directory. The installation program will automatically create this directory.

Another advantage of directories is that you can have files with the same name in different directories. This wouldn't be possible if you only worked with the root directory.

For example, you could have a directory containing tax information (TAXES90, TAXES91, etc.) with a file called MILEAGE in each directory. In the MILEAGE file you could keep track of your travel expenses for different years.

4.1 MS-DOS And Subdirectories

As we previously mentioned, each hard drive or diskette has a root directory. This root directory is always the highest level directory on every hard drive or diskette.

You can organize the files or the drive by creating new subdirectories of the root. These subdirectories can, themselves, contain other subdirectories, and so on. Here's a directory listing for a drive that contains two subdirectories called DOS and APPS as well as individual files. Notice that the listing is for the root directory of the C: drive - indicated by C:\.

```
 Volume in drive C is 386
 Volume Serial Number is 163B-4567
 Directory of C:\

DOS          <DIR>     10-12-90   1:18p
APPS         <DIR>     10-12-90   3:20p
AUTOEXEC BAT       294 02-03-91   8:36a
COMMAND  COM     47845 03-22-91   5:10a
CONFIG   SYS       122 02-15-91  11:41a
```

Directory structure

The above listing is an example of a directory structure of drive C:. At the top is the root directory. On the hard drive this directory contains only the most important system files, such as the AUTOEXEC.BAT, CONFIG.SYS and the COMMAND.COM command interpreter.

The example also shows two subdirectories: one containing the external DOS commands and programs of the operating system and another containing application programs. For example, this subdirectory might contain the database program dBASE IV and the word processing program Word.

Many users choose to store programs in a subdirectory separately from the data which they use. For example, this data might be stored in a subdirectory called C:\DATA\DB and C:\DATA\DOC.

You can use the MS-DOS Shell or the TREE command to display the structure of your hard drive.

Directory names

When assigning names to directories, remember that the names can't be longer than eight characters. The characters used for filenames are also valid for directory names.

All alphanumeric characters can be used. The letters from A to Z, the numerals 0 to 9 as well as the following special characters can be used for directory names:

!	exclamation point
#	number sign
$	dollar sign
%	percent sign
(	open parenthesis
)	close parenthesis
&	and
-	minus
_	underline

When lowercase letters are entered, DOS automatically converts them to uppercase letters.

The following characters cannot be used in directory names:

+	plus
=	equal
:	colon
;	semicolon
.	period
,	comma
*	asterisk
<	less than
>	greater than
/	slash
\	backslash

Imbedded spaces cannot be used because a space is used to separate parameters which would confuse DOS.

☞ The period character isn't entirely prohibited because it is used to separate the name and its extension.

In addition to these individual characters, some groups of characters are prohibited because they have special names (device names). Among these are: CON, AUX, COM1, COM2, PRN, LPT1, LPT2, LPT3, NUL, which are all device names.

Although directory names can also have extensions (e.g., DOCUMENT.DOC), they are rarely used with directories. There are various reasons why you shouldn't use extensions with directories. For example:

- You should ensure that a directory with an extension doesn't contain a file with the same name as the directory. Since files usually have an extension, you can avoid this problem by not adding an extension to a directory name.

- When you change to a directory you must always specify the complete name. So it's a lot easier to do this if you use shorter names (without extensions).

The directory names "." and ".."

When viewing the contents of a subdirectory, you'll also see two entries, which appear before the other subdirectories, that contain one and two periods. In its internal management of directories, MS-DOS manages these two additional directory names with the following meanings:

.	The period always refers to the current directory in MS-DOS. This entry is needed for the internal management of the directories.
..	The two periods always refer to the directory immediately above the current directory. MS-DOS also uses this entry for internal management.

☞ Later you'll see that you can use CD .. to move to the next highest directory.

Using directory names

If you have a lot of nested directories on your hard drive or diskette, you must be able to refer to them easily.

In MS-DOS, the entire name of any directory is composed of the path specification to the directory and the actual directory name. The path specification includes the names of directories to the specified directory, starting with the root directory. Directories are separated by a backslash character (\); path specifications can contain up to 63 characters.

For example, if the PERSONAL directory is a subdirectory of the TEXTS directory, the path specification would look like the following:

```
\TEXTS\PERSONAL
```

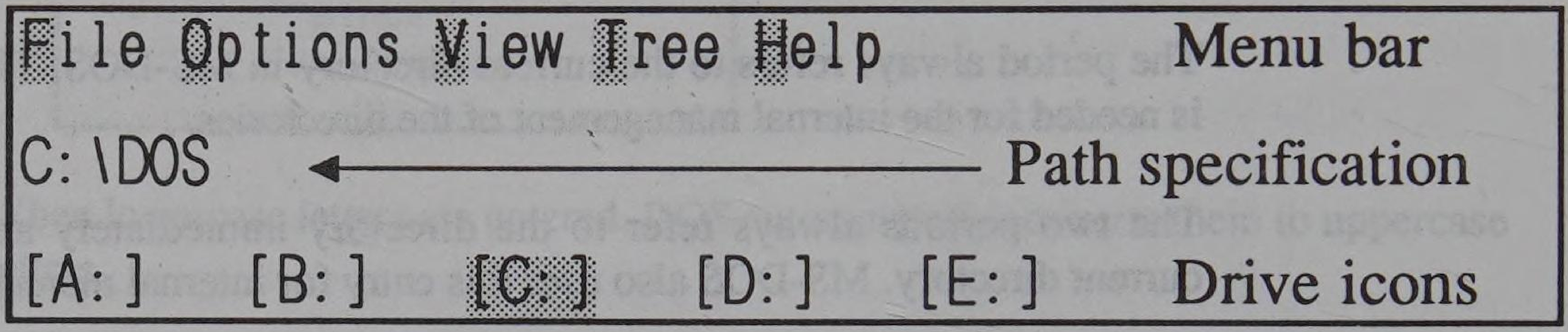
The backslash, which represents the root directory, is first. It's followed by the names of the directories down to the last directory. Each directory name is separated by a backslash. These specifications always refer to the current storage device (diskette or hard drive).

To refer to a directory from a different storage device as the current directory, specify the drive letter at the beginning of the path specification. This type of path specification looks like the following:

```
A:\TEXTS\LETTERS
```

Since you always identify drives with a colon, remember to enter a colon after the drive letter.

☞ In the MS-DOS Shell the pathname of the current directory is always specified in the line directly above the drive line. If you ever need a path specification but don't remember what it looks like, you can display it in the MS-DOS Shell.



The path specification in the MS-DOS Shell

The current directory

When MS-DOS works with directories, it manages a current directory for each drive. This is always the last directory you selected from the corresponding directory. As a rule, when you begin working at the computer, the current directory is the root directory.

☞ The MS-DOS Shell works a little differently. You must always define the drive to which your specifications refer. MS-DOS doesn't manage a current directory for every drive. So, regardless of the drive, the current directory in the MS-DOS Shell is always the root directory.

When changing drives, the current directory in the command interpreter is always reset, unless you specify another directory.

If the C:\DOS directory is the current directory, you can switch to drive A:, by typing:

```
A:Enter
```

To return to drive C:, type:

```
C:Enter
```

This switches you back to the C:\DOS directory.

Unless you specify otherwise, MS-DOS commands always affect the files in the current directory. To use a command on any other directory, you must specify the entire name of the directory for the parameters in the command.

Use this kind of directory management carefully. Be especially careful when editing data on a different storage device than the current one or transferring data between different storage devices (copying). For example, to copy all of the files on the current directory of the hard drive to a diskette, use the following command:

```
COPY *.* A:
```

MS-DOS copies the files into the current directory of the diskette instead of to the root directory. To copy to the root directory, the line should appear as follows:

```
COPY *.* A:\
```

To display the current directory of a storage device, use the CD command and specify the drive, whose current directory you want to display, after the command. For example, to display the current directory of drive A:, type the following:

```
CD A:
```

How many directories are possible?

Even though all the other directories can have an almost unlimited number of entries, the root directory has a limited number of entries. The reason for this is that the root directory always begins at the same disk location and has a fixed length which limits the number of entries that may follow. Subdirectories, on the other hand, can have an almost unlimited length, similar to files.

The root directory allows only a specific number of entries. Files and directories are both considered entries. The highest number of valid entries depends on the type of storage device. The following table shows the maximum number of entries possible in the root directory of different storage devices:

Storage device	Maximum number of entries
180K diskette	63
360K diskette	111
720K diskette	111
1.44 Meg diskette	223
1.2 Meg diskette	223
Hard drive	511

4.2 Setting Up A Directory

To create a new directory, use the MD (Make Directory) command. Type the name for the new directory after the command. Remember that directory names, similar to files, can only be eight characters long and that not all characters are allowed. Press Enter to confirm the name you've entered. To create a new directory, type the following:

```
MD directory_name
```

Creating a directory as a subdirectory in the current directory

To create a subdirectory, type the name of the new directory after MD. This new directory is always a subdirectory of the current directory. For example, if you're located in the C:\DOC directory and type the following command:

```
MD LETTERS
```

the LETTERS directory would be created as a subdirectory of the DOC directory. The pathname of the subdirectory is as follows:

```
C:\DOC\LETTERS
```

Creating a new subdirectory for any directory

To create a subdirectory for another directory besides the current one, simply specify the path of this directory to your new directory name. To create a subdirectory named PERSONAL for the DOC directory, from any directory, type the command line:

```
MD C:\DOC\PERSONAL
```

Remember that only one directory at a time can be created with a single command. If you wanted to create a sub-subdirectory for a subdirectory, the first subdirectory must already exist.

Suppose that you want to create a directory, named TEST1, that will be a subdirectory of the TEST directory. You would use the command:

```
MD C:\TEST\TEST1
```

This will only work if the TEST directory already exists. Again, you can only create one directory at a time with the MD command.

Directory names

MS-DOS manages directories similar to the way it manages files. So you cannot use a name for your subdirectory that has already been given to a file in the current directory. For example, you can't set up a subdirectory named PERSONAL in C:\DOC if this directory already contains a file named PERSONAL. However, you can avoid this problem by using extensions for files, and not for directories.

Also, remember which characters cannot be used in directory names.

Directories on diskettes

You can also set up directories on diskettes. However, it isn't as important to set up directories on diskettes as it is on hard drives because of the smaller number of possible files. To set up a directory on a diskette, either make the disk drive the current drive and create the directory with the following command:

```
MD directory_name
```

Or enter the command for creating a new directory directly from the current drive by adding the drive, where you want to create the directory, to the command:

```
MD A:\DOC
```

Automatic directories

Some programs automatically set up directories during installation. Usually when you install programs, you're prompted to enter the directory, in which you want to store the program files. If this directory doesn't exist on your storage device, the program automatically creates it during installation. Often programs also set up subdirectories for these directories. In these subdirectories, the installation programs place files that contain examples for the program or certain other necessary files. Some programs also set up directories for backup copies of files (e.g., Microsoft Works).

4.3 Choosing A Directory

When entering a command or working with a specific file, MS-DOS automatically looks for the command or file in the root directory if no other path is specified. If the command or file isn't in the current directory, MS-DOS displays the following error message:

```
Bad command or file name
```

In this case, make the directory containing the file the current directory. In the next few paragraphs we'll show you various ways to change directories.

Changing to any directory

To execute a command or find a file, make the directory, which contains the command or file, the current directory. Use the CD (Change Directory) command to do this. After the CD command, enter the name of the directory you want to make the current directory. The command line always has the following structure:

```
CD directory
```

You can also specify drives before specifying the actual directory name. For example, to change from the hard drive to the TEST directory, which is on the diskette in drive A:, use the following command line:

```
CD A:\TEST
```

To change from this directory to the LETTERS directory, which is a subdirectory of DOC located on hard drive C:, type the following command line:

```
CD C:\DOC\LETTERS
```

When changing to a directory on the current storage device, you don't have to specify the drive.

```
CD \DOC\PERSONAL
```

To change to any subdirectory (i.e., to make it the current directory), you must always specify the entire path.

Changing to a lower directory

To change to a directory that is a subdirectory of the current directory, simply specify the name of the subdirectory after the CD command.

If you're currently in the DOC directory and want to change to the PERSONAL subdirectory, simply type:

```
CD PERSONAL
```

When used this way, the CD command always moves you to the directory (if it exists) one level lower than the current directory.

Changing to a higher directory

The entire directory name doesn't have to be specified in order to move to a directory one level higher than the current one. Since MS-DOS always marks the higher directory with "..", you can type:

```
CD ..
```

☞ Remember that the parameter ".." is a result of MS-DOS setting up two special directories in each subdirectory, "." and "..". The first one "." represents the current directory while the second one ".." represents the next highest directory. So ".." can be used to move one level higher in the directory structure. The root directory doesn't have this special ".." directory because it's already the highest directory.

Changing to the root directory

Since MS-DOS always assigns the name "\" to the root directory, you can select the root directory of the current drive by typing:

```
CD \
```

☞ MS-DOS marks a current directory for every existing drive. When you change to another drive, the last directory you selected automatically becomes the current one.

You can't specify the root directory of a storage device by only using the drive label. You must specify the root directory with the drive label and a backslash symbol (\).

Displaying the current directory

To display the name of the current directory, type CD without using any additional parameters.

You can also display the current directories of other drives by typing the drive letter after the CD command:

```
CD A:
```

This information is especially important when working with data from a storage device other than the current one. For example, the following command displays the contents of the current directory of the diskette in A:

```
DIR A:
```

However, this isn't necessarily the contents of the root directory of A:. To display the contents of the root directory, type:

```
DIR A:\
```

Displaying the directory with PROMPT

To always display the current drive and directory on the screen, change the system prompt with the PROMPT command. To do this, type the following command line:

```
PROMPT $P$G
```

☞ When you install MS-DOS 5.0, this system prompt is automatically set up and included in the AUTOEXEC.BAT file.

$P represents the "current directory of the current drive" and $G represents the "greater than symbol" (>). The screen displays the current directory with a greater than symbol as the system prompt.

Other options are also available for changing the display of the system prompt using PROMPT. See “The MS-DOS 5.0 Commands” chapter for the options of the PROMPT command.

☞ Insert this command line into your AUTOEXEC.BAT file so that you don't have to type the PROMPT command to change your system prompt each time you start your computer.

4.4 Search Paths For Commands And Files

When you type a command or batch file name at the system prompt, MS-DOS searches for the command or batch file in the current directory. If the command or batch file isn't in the current directory, MS-DOS displays the error message:

```
Bad command or filename
```

You can, of course, precede the command or batch file name by a path if you know where these files are located, and MS-DOS can then execute them. But it becomes very tedious to have to type the entire pathname or change to a specific directory in order to execute a command. It would make life much easier if you were able to use these commands and batch files from any directory.

The PATH command provides such a way. When you define a search path with PATH, MS-DOS automatically searches the directories listed in the search path for commands and batch files.

Search paths for programs

MS-DOS considers all files with the extensions COM, EXE or BAT to be programs. The external MS-DOS commands are examples of these program files.

To call these commands and other program files from any directory, use the PATH command to define a search path.

☞ Frequently, the PATH command is inserted into the AUTOEXEC.BAT file to define the search path. By doing this, you won't have to re-enter the search path every time you boot your computer.

Creating a search path

To put a directory in the search path, enter the directory name after the PATH statement. It's important to enter the entire pathname. Enter the drive identifier as well as the directory name.

The command line for creating a search path looks like the following:

```
PATH directory_name
```

Since the program files that are used most often are usually in different directories, you can include several directories in the search path. Each directory name is separated by a semicolon.

There are two different ways of defining the search path:

```
PATH directory1;directory2;directory3
```

or

```
PATH=directory1;directory2;directory3
```

To add a directory to an existing search path, you must specify all of the previously defined directory names, in addition to the new one. If you don't the previously defined paths will no longer be valid.

In the following example we'll define a search path that makes the MS-DOS commands, the program files of Word and dBASE IV available.

You'll be able to call MS-DOS commands from any directory and start the Word and dBASE IV programs from any directory. The command line for defining this search path looks like this:

```
PATH = C:\DOS;C:\WORD;C:\DBASE4
```

Displaying the search path

If you type PATH without any parameters, the screen displays the previously defined search path:

```
PATH
```

☞ Some applications use OVL files, which are overlay files. These are parts of a program that aren't kept in memory at the same time. For these applications, the PATH command will not correctly search for the OVL files.

If this happens, use APPEND to create a search path for the OVL files. Most programs automatically prevent this kind of problem.

Sequence of search

When you type a command at the system prompt, MS-DOS first searches for it in the current directory. If you don't specify an extension, MS-DOS looks for files with the following extensions in this order: .COM, .EXE and .BAT.

If MS-DOS finds a COM file with the name you entered, it runs the program or command. Otherwise it searches for an EXE file and runs it if found. If it doesn't find one, it looks for a BAT file with the corresponding name, etc.

If there are two files in a directory with the same name, but with the extensions .COM and .BAT, MS-DOS will run the COM file.

☞ You can change the sequence of the search by entering the extension after the command. MS-DOS then looks for a command with the specified extension. To start a batch file named FORMAT.BAT in the directory of the MS-DOS commands instead of the MS-DOS FORMAT.COM command, specify the extension .BAT:

```
FORMAT.BAT
```

When MS-DOS searches beyond the current directory for files, it searches the directories in the sequence specified in the search path. This is why the MS-DOS directory is usually specified in the search path first. This enables you to find the MS-DOS commands quickly.

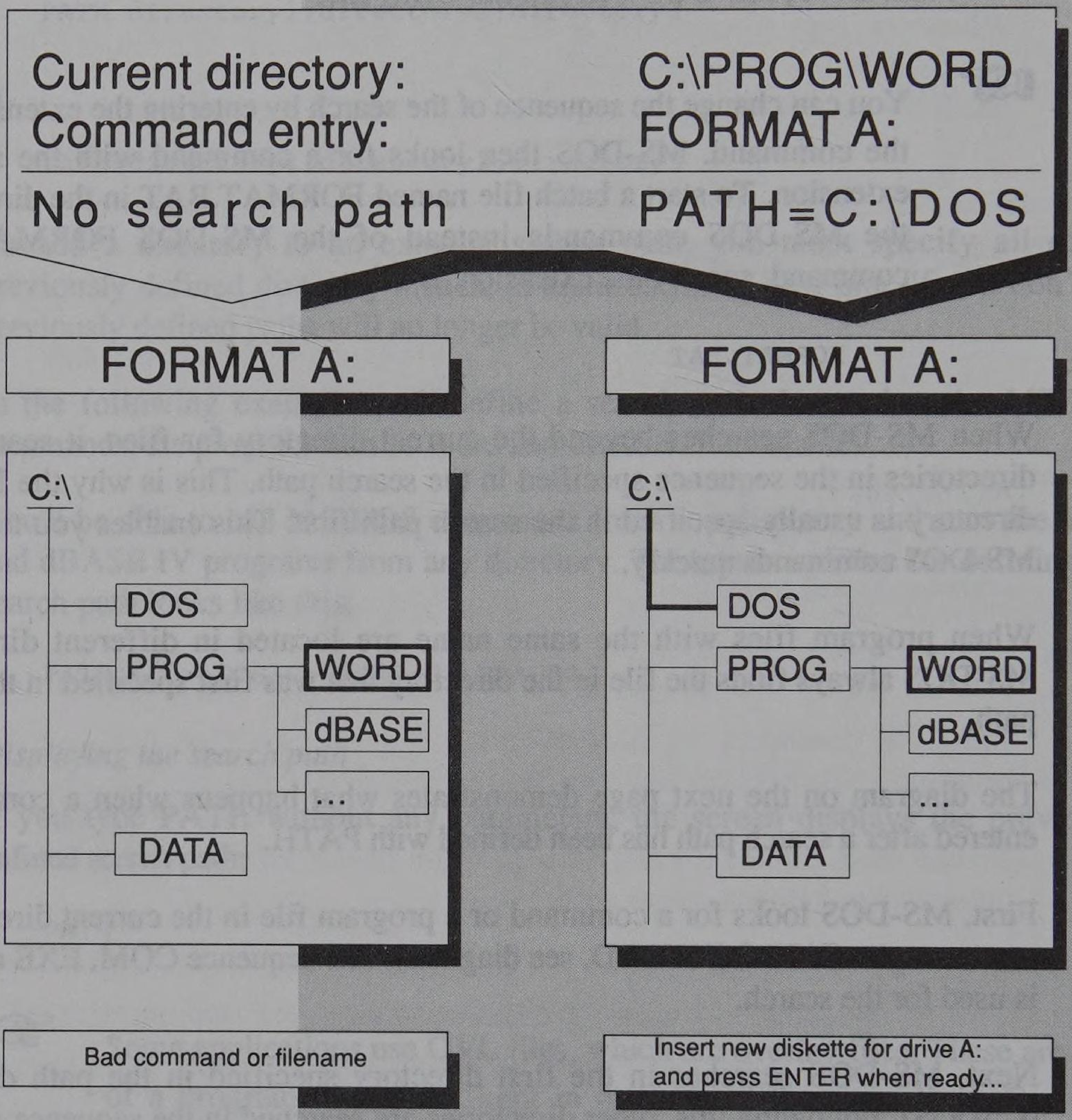
When program files with the same name are located in different directories, MS-DOS always finds the file in the directory that was first specified in the search path.

The diagram on the next page demonstrates what happens when a command is entered after a search path has been defined with PATH.

First, MS-DOS looks for a command or a program file in the current directory (in our example: C:\PROG\WORD, see diagram). The sequence COM, EXE and BAT is used for the search.

Next, MS-DOS searches in the first directory specified in the path definition (C:\DOS). Following this, other directories are searched in the sequence specified in the path.

The longer the path specification (i.e. the more directories that are specified in the search path), the longer it takes MS-DOS to search for the command you specify.


Current directory:
C:\PROG\WORD
Command entry:
FORMAT A:
No search path
PATH=C:\DOS
FORMAT A:
FORMAT A:
C:\
DOS
PROG
WORD
dBASE
...
DATA
C:\
DOS
PROG
WORD
dBASE
...
DATA

```
Bad command or filename
```

```
Insert new diskette for drive A:
and press ENTER when ready...
```

Meaning and function of PATH

Length of a search path

Defining a long search path will slow down MS-DOS. This happens because MS-DOS sequentially searches all of the directories in the search path in the specified sequence. So, it can take a while for MS-DOS to find the program you want, especially if it's in the last directory of the search path. If MS-DOS doesn't find the program, it displays the error message:

```
Bad command or filename
```

Which directories should be included in the search path?

As previously mentioned, you shouldn't include too many directories in the search path. In the next few sections we'll give you some hints on which directories to include in your search path and which ones to omit. We'll also discuss how you can start programs without including them in the search path.

It's not necessary to define a search path for directories with only one program file, such as directories with files for application programs. Usually it's better to create a batch file that calls the program.

For example, to start the Works program from any directory without specifying the C:\WORKS directory in the search path, use COPY CON, the EDITOR or a word processor to create the following batch file, named WORKS.BAT, in the DOS directory:

```
C:\WORKS\WORKS %1 %2 %3
```

☞ For more information about passing parameters, read the chapter on batch files.

If you create batch files this way but the programs aren't working properly, perhaps they shouldn't be included in the search path. Instead, try to solve this problem in a different way.

One option is to create another batch file that first changes the current directory to the directory containing the program files, and then calls the program.

For example, suppose that you want to be able to run the word processing program Word from any directory but don't want to include the Word directory in the search path. Create the following batch file in the DOS directory:

```
CD C:\WORD
WORD %1
```

You can also create a directory to store all of your batch files. This directory is then added to the search path ahead of the directory containing the MS-DOS commands. By doing this you can "replace" MS-DOS commands with batch commands using the same name. This lets you adapt MS-DOS commands to your own requirements.

For example, you could prevent MS-DOS from formatting your hard drive by creating a batch file, named FORMAT.BAT, that always calls FORMAT with the A: parameter. You must keep this file (FORMAT.BAT) in the BAT directory so that MS-DOS finds and runs it before it finds the FORMAT.COM program file in the DOS directory. Again, the BAT directory should be defined in the search path ahead of the MS-DOS commands directory.

Deleting search paths

To delete the defined search path, enter a semicolon after the MS-DOS PATH command:

```
PATH ;
```

☞ The space after the command is not absolutely necessary.

Remember that this command cancels the previously defined search path.

If you define a new search path that doesn't contain the directory of external (transient) MS-DOS commands, these commands will no longer be available for use. MS-DOS will display the error message:

```
Bad command or filename
```

Specify the complete path of MS-DOS commands or define a new search path that includes the directory of the commands.

Remember that every new definition of a search path cancels all previously defined search paths.

Search paths for files

Use the APPEND command to create, delete and display search paths for data files. This command applies only to files that aren't program files (files that don't have the extensions .COM, .EXE or .BAT).

You should only use APPEND when programs prompt you to do so. You may have trouble working with files found by APPEND. For example, instead of saving a file in the directory from which it was opened, APPEND saves it in the current directory. The old version of the file will still be in the original directory and the newer version will be in the current directory. Obviously this can lead to confusion and potential loss of data.

Creating a search path for data files

To define a directory as a search path for data files, specify it using the APPEND command. As with the PATH command, the drive can be specified as well as the entire directory name. There are two ways to define such a search path:

```
APPEND directory
```

or

```
APPEND=directory
```

To use APPEND to define a search path for several directories, specify the directory names separating each with a semicolon:

```
APPEND Directory1;Directory2;Directory3
```

or

```
APPEND=Directory1;Directory2;Directory3
```

The following is an example of a search path for two different text directories:

```
APPEND=C:\DOC\PERSONAL;C:\DOC\BUSINESS
```

Always define the complete search path when you use APPEND. Each time you call APPEND, the previous search paths are cancelled.

Deleting a search path

To delete a search path for data files type APPEND followed by a semicolon:

```
APPEND ;
```

☞ The space after the command is not absolutely necessary.

Displaying a search path

Typing APPEND without parameters displays the previously defined search path:

```
APPEND
```

Using APPEND for the first time

The first time you use the APPEND command, you can use the /E switch. This creates an environment variable name APPEND in which the appended search path is stored. Saving the appended search path in an environment variable has two advantages: you can extend the appended search path with a batch file and a program can easily determine the appended search path.

Search paths for both commands and data files

One form of the APPEND command can be used to define a search path for commands as well as data files. This form of the APPEND command is a substitute for the PATH command.

Defining a common search path for programs and data files

Use the APPEND command's /X:ON switch to define a search path for both commands and data files. For example, if your word processing program and the document files that you edit with the word processor are found in a directory named WP, use this command:

```
APPEND C:\WP /X:ON
```

Then, both the word processing program and the documents can be accessed from any directory.

To turn off the command search`path, use the /X:OFF switch. When you turn off the command search, APPEND continues to use the search path for opening data files.

Finding files regardless of the path specification

Let's say that you are using an application to process a file named LETTER.TXT that you believe is in the C:\DOC directory. You might specify the filename preceded by the path: C:\DOC\LETTER.TXT. If you are mistaken and LETTER.TXT is in the DATA directory, not in the DOC directory, then your application will not find your file.

If you use the APPEND command with the /PATH:ON switch, MS-DOS will first look for the DOC directory for your file. If it isn't there it will look for the file in the appended search path. For example, if this command were used:

```
APPEND C:\DATA /PATH:ON
```

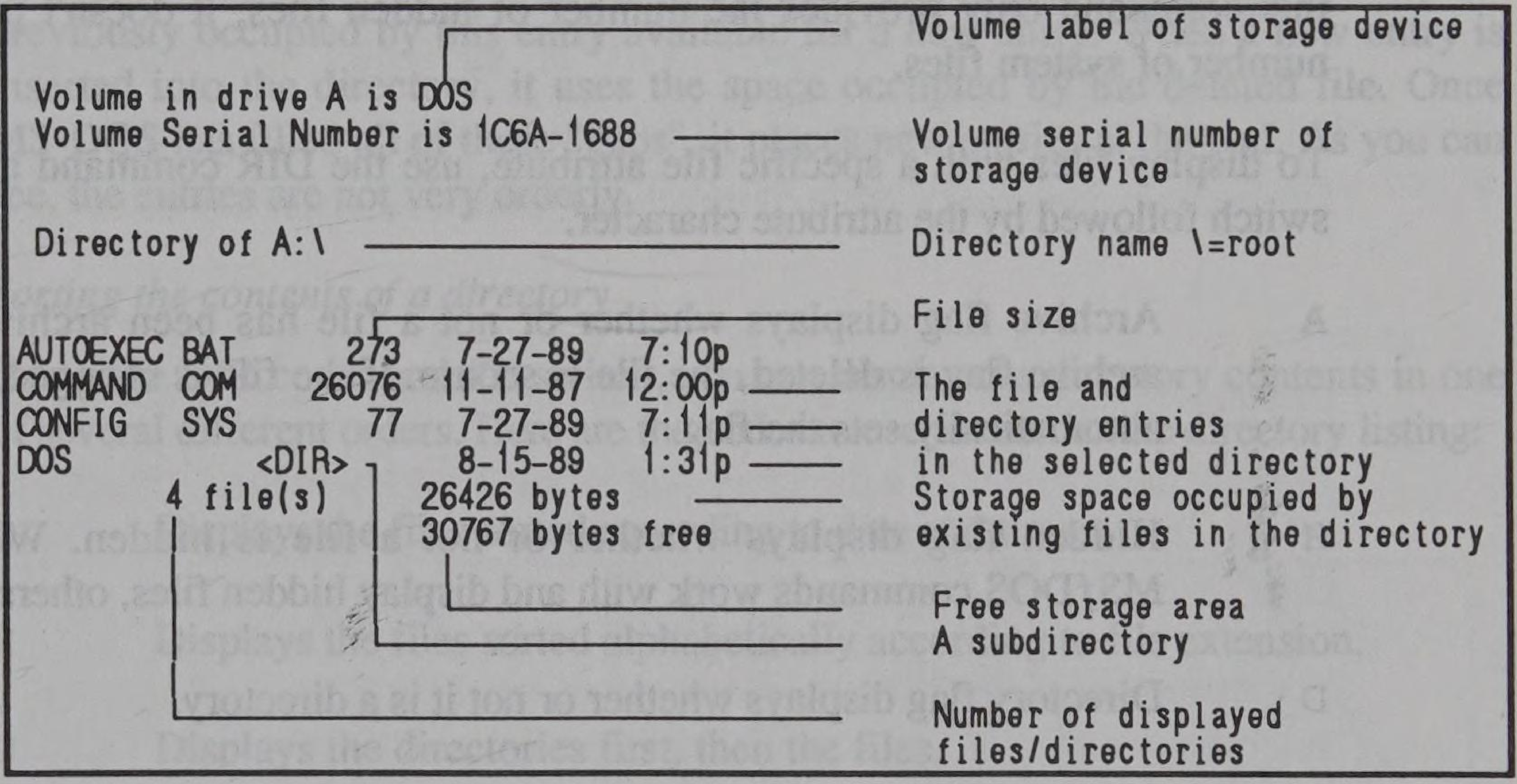
then the above application would be able to find the LETTER.DOC data file.

Use the /PATH:OFF switch to cancel this option.

4.5 Displaying And Printing The Directory

Use the DIR command to display the contents of directories. This will provide information about the directory.

Regardless of how DIR is started, the contents of the directory are always displayed. This directory has the following structure and provides the following information:



```
 Volume in drive A is DOS
 Volume Serial Number is 1C6A-1688

 Directory of A:\

AUTOEXEC BAT       273   7-27-89   7:10p
COMMAND  COM     26076  11-11-87  12:00p
CONFIG   SYS        77   7-27-89   7:11p
DOS          <DIR>       8-15-89   1:31p
        4 file(s)     26426 bytes
                      30767 bytes free
```

Volume label of storage device
Volume serial number of storage device
Directory name \=root
File size
The file and directory entries in the selected directory
Storage space occupied by existing files in the directory
Free storage area
A subdirectory
Number of displayed files/directories

Directory contents

MS-DOS provides the following information about the storage device and the directory:

Volume label of storage device
: 11 character volume label from FORMAT or LABEL.

Volume serial number of storage device
: 8 digit hexadecimal number used by MS-DOS to determine when a diskette is changed.

Directory
: Path specification of the displayed directory.

Entries Filename, file size, creation date, creation time.

Number files
: Number of entries in this directory listing.

Storage space occupied by files
: The amount of storage space occupied by all files in this directory listing.

Unused storage area

The amount of unused storage space remaining on the storage device.

Displaying files with certain archiving attributes

If a directory contains hidden or system files, they are not normally displayed in a directory listing.

To see if a storage device has hidden files, use the CHKDSK command. However, this command only provides the number of hidden files; it doesn't provide the number of system files.

To display files with a specific file attribute, use the DIR command and the /A: switch followed by the attribute character.

`A`	Archive flag displays whether or not a file has been archived. If the archive flag is deleted, the file is secure. If the file is changed, MS DOS automatically sets the flag.
`H`	Hidden flag displays whether or not a file is hidden. While some MSƒDOS commands work with and display hidden files, others do not.
`D`	Directory flag displays whether or not it is a directory.
`R`	Read-Only flag displays whether or not a file is write protected.
`S`	System flag displays whether or not a file is a system file. Some special programs treat system files differently (for example, the hard drive optimizer SD.EXE of Norton Utilities).

You can also precede the file attribute with a minus sign to display only the files that don't have the indicated attribute.

Example

To display all files that don't have the archive flag set, type:

```
DIR /A:-A
```

Use the following command line to display only hidden system files in the root directory of the hard drive:

```
DIR C:\ /A:S
```

☞ You can also display the various file attributes in the MS-DOS Shell. For more information on how to do this, refer to the chapter on the MS-DOS Shell.

Display order

DIR normally displays the files and directories in the order in which they were created. If you delete an entry in this directory, MS-DOS considers the space previously occupied by this entry available for a new entry. When a new entry is inserted into the directory, it uses the space occupied by the deleted file. Once MS-DOS has filled all of these "gaps", it places new entries at the end. As you can see, the entries are not very orderly.

Sorting the contents of a directory

Using the DIR command /O: switch, you can display the directory contents in one of several different orders. Here are the codes to use to reorder the directory listing:

D	Displays the files sorted according to date and time.
E	Displays the files sorted alphabetically according to file extension.
G	Displays the directories first, then the files.
N	Displays the files alphabetically sorted according to the filenames.
S	Displays the files sorted according to the size of the file.

To reverse the sorting sequence, precede the code with a minus sign.

Example

In your C:\DOC directory you're looking for the last text file that was created. However, you can't remember the filename. Use this command line:

```
DIR C:\DOC\*.DOC /O:D
```

You'll see a listing of all of the DOC files, sorted by date, with the last edited file listed last.

Displaying the contents of the current directory

Display the contents of the current directory by typing DIR. All of the information about the current directory will appear.

Displaying the contents of any directory

To display the contents of any directory, enter the name of the directory after the DIR command. If the directory is on a different storage device than the current one, specify the drive identifier (letter):

```
DIR Directory_name
```

Use the following command line to display the contents of the DOS directory from any directory on your hard drive:

```
DIR \DOS
```

Display the contents of the root directory of drive A: from any drive by typing:

```
DIR A:\
```

Contents of a subordinate directory

Display the contents of a directory that is one level lower than (directly subordinate to) the current directory by entering the name of this directory after DIR. You don't have to specify the entire pathname.

For example, to examine the contents of the PERSONAL directory, which is a subdirectory of DOC, type the following command line from the DOC directory:

```
DIR PERSONAL
```

Contents of the "." and ".." directories

Each subdirectory contains two additional directory entries named "." and "..", which MS-DOS requires for directory management. The "." represents the current directory and the ".." represents the next highest directory. Test this from within a subdirectory by typing both of the following command lines one after the other:

```
DIR .
DIR ..
```

The first command line displays the same information you would have seen if you had entered only DIR (the current directory). The second command line displays the contents of the next highest directory. Use exactly the same parameter to change to the next highest directory.

```
CD..
```

Displaying the directory contents one page at a time

When you use the DIR command and there are a lot of entries in a directory, the contents of the directory might scroll by faster than you can read them. This happens because there are more files than will fit on the screen at the same time. Only the last files in the directory will remain on the screen.

To prevent the screen from scrolling by, use the /P switch. MS-DOS then displays one screen page of the directory at a time and waits for a key press before displaying the next page. The command line is as follows:

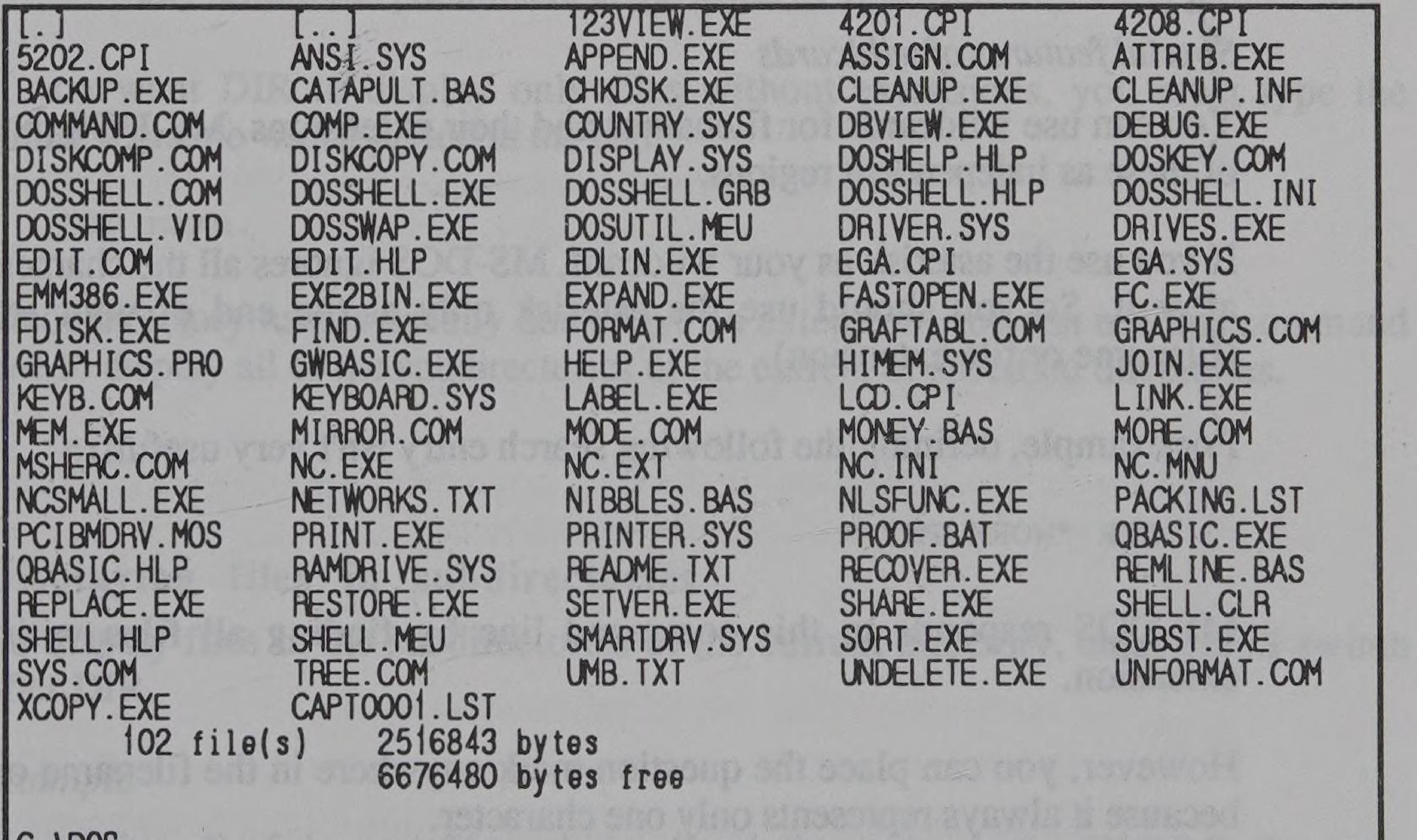
```
DIR Directory /P
```

Displaying the contents in five columns

Occasionally you may want to display the entries of a directory on a single screen so that you can get a general overview of the contents. MS-DOS provides a way to do this.

Use DIR's /W switch. This lists the entries on the screen in five columns. To display the directory in columns, use the following command line:

```
DIR Directory /W
```



```
[.]            [..]           123VIEW.EXE    4201.CPI       4208.CPI
5202.CPI       ANSI.SYS       APPEND.EXE     ASSIGN.COM     ATTRIB.EXE
BACKUP.EXE     CATAPULT.BAS   CHKDSK.EXE     CLEANUP.EXE    CLEANUP.INF
COMMAND.COM    COMP.EXE       COUNTRY.SYS    DBVIEW.EXE     DEBUG.EXE
DISKCOMP.COM   DISKCOPY.COM   DISPLAY.SYS    DOSHELP.HLP    DOSKEY.COM
DOSSHELL.COM   DOSSHELL.EXE   DOSSHELL.GRB   DOSSHELL.HLP   DOSSHELL.INI
DOSSHELL.VID   DOSSWAP.EXE    DOSUTIL.MEU    DRIVER.SYS     DRIVES.EXE
EDIT.COM       EDIT.HLP       EDLIN.EXE      EGA.CPI        EGA.SYS
EMM386.EXE     EXE2BIN.EXE    EXPAND.EXE     FASTOPEN.EXE   FC.EXE
FDISK.EXE      FIND.EXE       FORMAT.COM     GRAFTABL.COM   GRAPHICS.COM
GRAPHICS.PRO   GWBASIC.EXE    HELP.EXE       HIMEM.SYS      JOIN.EXE
KEYB.COM       KEYBOARD.SYS   LABEL.EXE      LCD.CPI        LINK.EXE
MEM.EXE        MIRROR.COM     MODE.COM       MONEY.BAS      MORE.COM
MSHERC.COM     NC.EXE         NC.EXT         NC.INI         NC.MNU
NCSMALL.EXE    NETWORKS.TXT   NIBBLES.BAS    NLSFUNC.EXE    PACKING.LST
PCIBMDRV.MOS   PRINT.EXE      PRINTER.SYS    PROOF.BAT      QBASIC.EXE
QBASIC.HLP     RAMDRIVE.SYS   README.TXT     RECOVER.EXE    REMLINE.BAS
REPLACE.EXE    RESTORE.EXE    SETVER.EXE     SHARE.EXE      SHELL.CLR
SHELL.HLP      SHELL.MEU      SMARTDRV.SYS   SORT.EXE       SUBST.EXE
SYS.COM        TREE.COM       UMB.TXT        UNDELETE.EXE   UNFORMAT.COM
XCOPY.EXE      CAPT0001.LST
      102 file(s)   2516843 bytes
                    6676480 bytes free

C:\DOS>
```

Contents of directory displayed in 5 columns by DIR/W

Contents of directory without additional information

To display only the names of your files and directories, use the /B switch. When this switch is used, no other information is displayed on the screen.

Selecting files with substitute characters

DIR is suitable for checking whether specific files are in specific directories or for searching for files whose name you've forgotten.

For example, suppose that you wrote a letter with a word processor but you can't remember the name you saved the letter under. All you can remember is that the filename starts with "B". You could find the file by displaying the directory in which you saved it, but it would take a lot of time because there are so many files to check. It would be a lot faster to display only the files that start with "B". In order to do this, MS-DOS provides the wildcard.

Wildcards are characters that can represent any character or character sequence within a filename.

MS-DOS provides two wildcards (placeholders) that can be used to select certain files:

*	Represents any remainder of a filename or extension.
?	Represents a single character in a filename or its extension.

Special features of wildcards

You can use wildcards for filenames and their extensions. MS-DOS manages both of these as independent regions.

If you use the asterisk as your wildcard, MS-DOS ignores all the characters entered after it. So you should use the asterisk only at the end of your search entry (filename or file extension).

For example, defining the following search entry isn't very useful:

```
DIR *WORK.TXT
```

MS-DOS responds to this command line by finding all files with the .TXT extension.

However, you can place the question mark anywhere in the filename or extension because it always represents only one character.

When you use the asterisk or the question mark at the end of the filename or extension, it can also represent a single character that is missing.

The following are a few examples of how you can utilize wildcards.

If you would like to display all files that start with the characters LETTER, use the following command line:

```
DIR LETTER*.*
```

The screen displays all files with the filename LETTER. In this case the wildcard can also represent a nonexistent character. This always occurs when you specify the wildcard as the last character in the filename or extension. So it doesn't matter if you use "*" or "?".

To display all files with the .BAT file extension that begin with the letter "B", use the command line:

```
DIR B*.BAT
```

You could use one of the two following command lines to display all of the files in the C:\BAT directory with either the .BAT or .BAK extension:

```
DIR C:\BAT\*.BA?
DIR C:\BAT\*.BA*
```

☞ The DIR command treats filenames and wildcards somewhat differently than the other MS-DOS commands. If the actual filename or extension isn't present, DIR automatically uses the asterisk "*". So DIR without any additional parameters is the same as DIR *.*.

If you want DIR to display only files without extensions, you must type the actual filename with a period at the end:

```
DIR name.
```

Since directory names usually don't have an extension, you can use this command line to display all of the subdirectories of the current or specified directories.

```
DIR *.
```

Displaying files in subdirectories

To display files in the subdirectories of the current directory, enter the /S switch after DIR.

Example

To display all of the text files within all subdirectories in the C:\DOC directory, use the following command line:

```
DIR C:\DOC\*.TXT /S
```

You can use this variation to search for a specific file anywhere on your hard drive. For example, if you want to know in which directory the file BOOK.DOC is located, type the following command from the root directory of the hard drive:

```
DIR BOOK.DOC /S
```

The screen will display all directories that contain this file.

Defining DIR default switches

You can specify switches for the DIR command to use by default if they aren't overridden. The DIRCMD environment variable is used to set these defaults. For example, if you always want to view the directory contents a page at a time type:

```
SET DIRCMD=/P
```

Now typing only DIR actually defaults to DIR /P since the DIRCMD environment variable contains the switches for the DIR command.

To display the directory sorted by date, set the DIRCMD environment variable by using the following command line:

```
SET DIRCMD=/O:D
```

Wildcards can also be used to determine which files to display. If you want the DIR command to display only text files with the .TXT extension, set the environment variable DIRCMD by using the following:

```
SET DIRCMD=*.TXT
```

☞ When setting this variable, remember that DIR will only display files specified by the switches. If the DIRCMD variable was set as described above and you display the contents of a directory with DIR before calling the DELETE command, not all of the files in the directory will be displayed. You may accidentally delete files.

To remove this definition of the DIR command, specify the DIRCMD variable without any definition, similar to the following:

```
SET DIRCMD=
```

☞ You can define this environment variable in the AUTOEXEC.BAT file. For example, you may want to display the directory by page by page.

Printing a directory

Occasionally you might want a printout of a directory. This means that you must send the contents of the directory to an attached printer.

To print a directory, simply redirect the output of the DIR command to your printer. Use the following command line for LPT1 and substitute the appropriate parameter for your printer:

```
DIR >LPT1
```

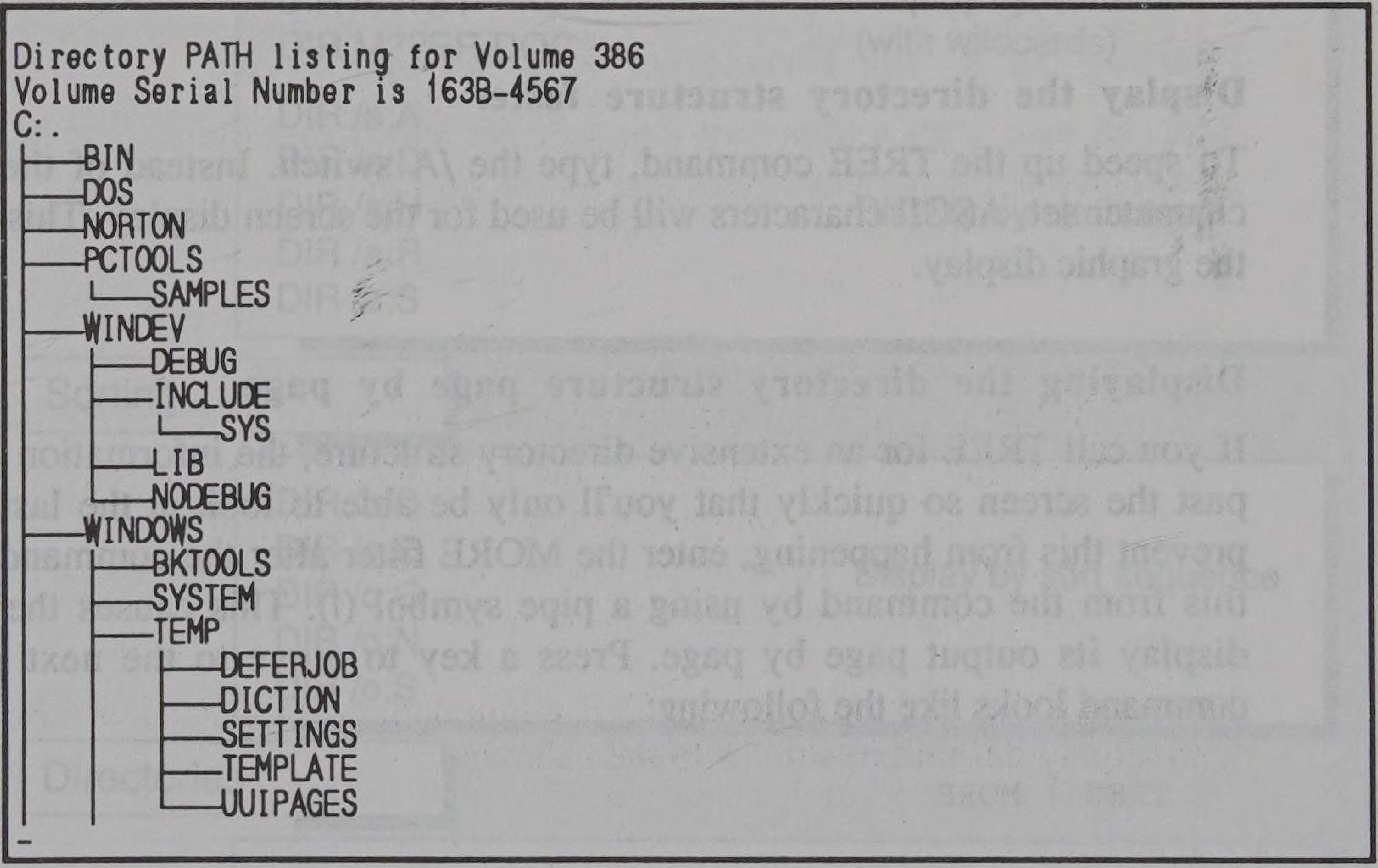
4.6 Displaying And Printing All Directories

The TREE command displays a graphic overview of the directories on a storage device.

Displaying all of the directories of the current storage device

The TREE command can also be used to display all of the directories of the current storage device. Remember that this command only displays directories and their subdirectories, not the files contained within. Subdirectories are always indented so that you can distinguish them from higher level directories. To display the structure of the current storage device, type the following:

```
TREE \
```



```
Directory PATH listing for Volume 386
Volume Serial Number is 163B-4567
C:.
├───BIN
├───DOS
├───NORTON
├───PCTOOLS
│   └───SAMPLES
├───WINDEV
│   ├───DEBUG
│   ├───INCLUDE
│   │   └───SYS
│   ├───LIB
│   └───NODEBUG
├───WINDOWS
│   ├───BKTOOLS
│   ├───SYSTEM
│   ├───TEMP
│   │   ├───DEFERJOB
│   │   ├───DICTION
│   │   ├───SETTINGS
│   │   ├───TEMPLATE
│   │   └───UUIPAGES
│   │
_
```

An example listing of the TREE command

Displaying all of the directories of any storage device

To display the structure of a different drive than the current one, specify this after the command. The command line would look like this:

```
TREE drive
```

TREE then displays only the directories beneath the current directory of the specified drive. To display all directories, enter the backslash for the root directory, for example:

```
TREE C:\
```

The directory structure of any subdirectory

If you enter a directory after TREE, the structure of this directory will be displayed. You'll see an overview of the existing subdirectories.

If you have a subdirectory named DOC and want to display its structure, type the following command line:

```
TREE \DOC
```

Displaying directories and files

If you want to display the files of these directories, type the /F switch after TREE. For the DOC directory your entry would look as follows:

```
TREE \DOC /F
```

Display the directory structure faster

To speed up the TREE command, type the /A switch. Instead of the extended character set, ASCII characters will be used for the screen display. This simplifies the graphic display.

Displaying the directory structure page by page

If you call TREE for an extensive directory structure, the information will scroll past the screen so quickly that you'll only be able to look at the last page. To prevent this from happening, enter the MORE filter after the command. Separate this from the command by using a pipe symbol (|). This causes the screen to display its output page by page. Press a key to move to the next page. The command looks like the following:

```
TREE | MORE
```

☞ To create the pipe symbol (|) on international keyboards, press:

Alt + 1 2 4

To display all of the directories and files of drive C: page by page, use the following command line:

```
TREE C:\ | MORE /F
```

Basic form of DIR

DIR corresponds to DIR *.*	Display all (non-hidden) files

Display modes

DIR /P	Page display
DIR /W	Wide display

Selection

DIR A*.TXT DIR M??ER.DOC	Display by filenames (with wildcards)
DIR /a:A DIR /a:D DIR /a:H DIR /a:R DIR /a:S	Display by attributes

Sorting

DIR /o:D DIR /o:E DIR /o:G DIR /o:N DIR /o:S	Display by sort sequence

Directories

DIR TEXT DIR C:\TEXT DIR A: DIR A:\SYSTEM	Display other directories or another storage medium
DIR ..	Display higher directories

Directory tree

DIR .... /S	Display all higher directories

Different methods of displaying the directory

Redirecting output of the directory structure to a file

If you want to place comments next to the directory structure and then print, collect the displayed directories in a file and edit them with a word processing program. Use the following command line:

```
TREE Drive >Dir.txt
```

In our example, the information is redirected to the DIR.TXT file of the current directory.

☞ When you redirect the screen output to a file, don't use the MORE filter. You won't be able to see anything on the screen but you will still have to press Enter several times before the MS-DOS system prompt returns.

Printing all of the directories of a storage device

You can also print a directory structure by redirecting the screen output to the printer. All of the options mentioned for TREE can be used:

```
TREE >LPT1
```

For LPT1, enter the appropriate parameter for your printer.

4.7 Renaming Directories

You cannot simply use the RENAME command to rename directories because this command only applies to files. Several steps are needed in order to rename a directory:

☞ It's much easier to rename directories with the MS-DOS Shell.

Renaming a directory that doesn't have subdirectories

Make a directory with the new name:

```
MD Newdir
```

Copy all of the files from the previous directory into the new directory:

```
COPY Prev_Dir\*.* Newdir
```

Now delete the files in the old directory:

```
DEL Prev_Dir\*.*
```

Answer the security prompt with Y.

Finally, remove the empty subdirectory:

```
RD Prev_Dir
```

☞ The COPY command doesn't copy hidden and system files. This means that you cannot delete the old directories because you can't use DEL to delete hidden, system and write protected files. First use the ATTRIB command to change the file's attributes. Then copy the files to the new directory and change the attributes back to their original state. Finally delete these files from the previous directory.

Renaming a directory with subdirectories

If the directory you want to rename has subdirectories, use the XCOPY command to copy instead of COPY. This command can also copy subdirectories.

Make your new directory:

```
MD Newdir
```

Copy the files from the old directory into the new directory:

```
XCOPY Prev_Dir\*.* Newdir /S
```

The /S switch is very important because it helps you copy the subdirectories along with the files. Delete all of the files in the subdirectories with DEL, then remove the subdirectories with RD. Finally, use CD.. to move one directory level higher and remove the directory:

```
RD Prev_Dir
```

☞ The XCOPY command of DOS 5.0 doesn't copy hidden and system files. This means that you cannot delete the old directories because you can't use DEL to delete hidden, system and write protected files. First use the ATTRIB command to change the file's attributes. Then copy the files to the new directory and change the attributes back to their original state. Finally delete these files from the previous directory.

Renaming directories with the MS-DOS Shell

It's considerably easier to rename a directory with the MS-DOS Shell. Mark the directory you want to rename by placing the cursor on this directory and pressing Enter.

Then press Alt to go to the menu and use the cursor keys and Enter to choose the **File** menu. Once there, choose the **Rename** command and press Enter.

Enter your new name for the directory in the dialog box after "New Name" and confirm by pressing Enter. It doesn't matter whether or not the directory has subdirectories.

4.8 Removing Directories

To delete a directory from the hard drive or diskette, use the RD command. Enter the name of the directory to delete after the command and press Enter. The command line for deleting a directory must have the following structure:

```
RD Directory_name
```

Remember that you can only delete directories when they are empty (i.e., when they no longer contain any files or subdirectories). MS-DOS considers a directory to be empty when it only contains the entries "." and "..".

To remove a subdirectory of the current directory, specify the name of the subdirectory after the RD command. If you want to delete a directory from any other directory, specify the entire path of the directory to be deleted, similar to setting up subdirectories. The command line should look as follows:

```
RD C:\DOC\BUSINESS
```

Use this line to delete the BUSINESS subdirectory from the DOC directory. Remember that the directory must be empty before you can delete it.

☞ You cannot remove a directory if it's the current directory or if you used SUBST to redirect a drive to this directory.

What do I do if the directory isn't empty?

If you try to delete a directory but an error message appears on the screen, the directory probably isn't empty. Display the contents of the directory with:

```
DIR Directory_name
```

If there are still files in the directory, make it the current directory and delete the existing files with the following command:

```
DEL *.*
```

If the directory you want to delete still has subdirectories, remove them with RD. After deleting the existing files and directories, use DIR to look at the contents of the directory. If you still see files displayed on the screen, these files are write protected.

Deleting write protected files

Write protected files have the Read-Only file attribute assigned to them. So DEL cannot be used to delete these files. In order to delete these files, remove the

Read-Only file attribute with the ATTRIB command. Call the command, specify -R (for remove Read-Only file attribute) and then the name of the file. The command line should look as follows:

```
ATTRIB -R TEST.TXT
```

After removing the Read-Only file attribute, use DEL to delete the file.

The directory seems to be empty

When you use DIR to display the directory, it no longer shows any files in the directory. If you still can't delete it, then it only seems to be empty. There are probably still some hidden files or files with the system file attribute.

To test this, use DIR to display all hidden and system files. Type:

```
DIR *.* /A:H
DIR *.* /A:S
```

If the screen displays files, change their file attributes. Use the ATTRIB command to do this. The command lines for removing the file attributes look like the following:

For hidden files:

```
ATTRIB -H filename
```

For system files:

```
ATTRIB -S filename
```

You can also use the MS-DOS Shell to check for hidden or system files. The MS-DOS Shell lists hidden and system files when it displays the contents of a directory. Look at the file attributes by calling the **File Display Options** command, and activating the **Display hidden/system files** command in the **Options** menu. Use the **Show Information** command from the **Options** menu to display the attributes of every marked file.

Hidden files have the file attribute "H", system files have "S". To remove these file attributes, use the **Change Attributes** command from the **File** menu.

After removing the file attributes, use DEL to delete the files. The directory must be empty. Make the next highest directory the current one and delete the subordinate directory with the following command:

```
RD Directory name
```

4.9 Turning A Directory Into A Pseudo Drive

The SUBST command is used to substitute a drive identifier for a directory. Rather than referencing to a directory, you can use the drive letter to refer to the contents of the directory.

This method will allow you to use programs that are set up for a system with two disk drives and adapt them to a hard drive system with only one disk drive. The second drive (which really doesn't exist) is created by turning a directory into a pseudo drive.

Defining the assignment

Use the command:

```
SUBST drive directory
```

to redirect each access from the drive to the specified directory.

To use this command, MS-DOS must recognize the drive identifier as a valid one. If necessary, use the LASTDRIVE command to increase the number of drives possible in the CONFIG.SYS file.

Here's an example:

```
SUBST B: C:\EXCEL\XLS
```

In this example, drive identifier B: is defined as an alternative way of referencing the contents of the directory C:\EXCEL\XLS. Now the following two commands produce the same results:

```
DIR C:\EXCEL\XLS
```

and

```
DIR B:
```

In other words, all references to B: are actually redirected to C:\EXCEL\XLS.

By defining directories as pseudo drives, you can avoid entering awkward path specifications. For example, if you don't have a drive D: and you often work with the C:\DATA\DOC\PERSONAL directory, then you can make this directory into a pseudo drive:

```
SUBST D: C:\DATA\DOC\PERSONAL
```

To copy a file named LETTER1.TXT from drive A: to the PERSONAL directory, simply type this short command line:

```
COPY A:\LETTER1.TXT D:
```

Removing the assignment

To remove an assignment, specify the redirected drive with the /D switch:

```
SUBST Drive /D
```

You won't be able to do this if the redirected drive is the current drive in MS-DOS. You also shouldn't cancel the assignment as long as application programs are working with this drive.

Example

For our pseudo drive D:, the following command line is used to cancel the assignment:

```
SUBST D: /D
```

Displaying the assignment

Entering SUBST without any parameters displays all redirected drives with their directories:

```
SUBST
```

Displaying the actual name of the drive

You can use the TRUENAME command to display the real name (the original directory name) of the pseudo drive.

To do this, simply type the command along with the drive whose true name you want to display:

```
TRUENAME drive
```

The screen then displays the correct name of the directory.

If you redirected the C:\TEMP directory to D:, type the following:

```
TRUENAME D:
```

The screen then displays the original name of the pseudo drive:

```
C:\TEMP
```

You can always use the TRUENAME command to check whether or not drives are redirected. This command always displays the original name.

Chapter 5

Handling Files

Files are groups of data that belong together. A file may be a program file, a group of commands, a text document, graphical or numerical data. A filename is used to identify the data.

5.1 Naming Files

Every file must have a name. In this section we'll present some guidelines for assigning filenames.

Possible filenames

A filename consists of a name, which can contain up to eight characters, and an extension. The filename extension can contain up to three characters.

All alphanumeric characters (letters from A to Z, the numbers from 0 to 9 as well as some special characters) can be used in filenames.

`!`	`HELLO!.TXT`
`#`	`TEXT#.TXT`
`$`	`SALARY$.DBF`
`%`	`TAXES%.DBF`
`(`	`(TEST.TXT`
`)`	`(TEST).WKS`
`&`	`90&91.DBF`
`-`	`LETTER-1.TXT`
`_`	`MEYER_90.TXT`

Lowercase letters are automatically converted to uppercase letters.

The following characters cannot be used in filenames:

+	plus
=	equal
:	colon
;	semicolon
.	period
,	comma
*	asterisk
<	less than
>	greater than
/	slash
\	backslash

The space character also cannot be used because it is used as a separator between elements of commands (parameters). For example, FORMAT A: /4.

☞ The period separates the filename from its extension.

Along with these single characters, there are also groups of characters which cannot be used for filenames. Among these are: CON, AUX, COM1, COM2, PRN, LPT1, LPT2, LPT3, NUL. All of these names are significant to DOS because they are names of devices.

The file extension

Certain file extensions have a fixed meaning in MS-DOS. The most important extensions are:

.SYS	Used for system files.
.COM	Used for MS-DOS command files. COM files are linked to a specific starting address and are generally smaller than comparable EXE files.
.EXE	Used for MS-DOS executable files or application programs.
.BAT	Used for batch files.
.BAK	Used for backup copies that are automatically created by text editors and some word processors.

Other extensions are commonly used as a result of being assigned by significant user programs, such as Word, WordPerfect and WinWord. For example:

.TXT	For text files (in EDIT).
.DOC	For document files (Word, WordPerfect or WinWord).
.DBF	dBASE database files.

.BAS	For BASIC programs.
.PAS	Turbo Pascal files.
.TMP	Used for temporary files, which are used by application programs to store data briefly. Usually, temporary files are automatically deleted. However, if a program isn't ended properly (e.g., you switch off the computer while you're still working with a program), these temporary files aren't deleted and occupy disk space.
.WKS	Lotus 1-2-3 file spreadsheet files.
.WK1	Lotus 1-2-3 file spreadsheet files.
.XLS	Excel spreadsheet files.

Specifying more than one file using wildcards

To address more than one file with an MS-DOS command, wildcards can be used. Use wildcards to specify one or more characters:

*	Replaces all characters following the asterisk in filenames or extensions.
?	Replaces any single character in a filename or its extension. If the question mark is placed at the end of a filename, it can represent any character or no character.

The "*" character in filenames or extensions determines that the rest of the characters in the filename or extension can be any character. So if you enter another character after "*", it will be ignored.

Examples

```
A*.EXE
```

The example above specifies all files whose names start with A and have the .EXE extension (APPEND.EXE, ATTRIB.EXE).

```
M??ER.TXT
```

This example applies to all files beginning with M and ending with ER that have 5 letters in the filename and the .TXT extension (MEIER.TXT MAYER.TXT, MEYER.TXT).

```
*.*
```

The above example specifies all files.

```
F*T.COM
```

The previous example refers to all files beginning with F that have the .COM extension. The character T is ignored because it follows the "*" wildcard. This example could be used to address FORMAT.COM, FDISK.COM.

```
S???.*
```

This example specifies all files with names that begin with S and have no more than four characters. The extension could be anything (SORT.EXE or SYS.COM).

Wildcards can only be used with existing files. If you're creating a new file, you won't be able to use "*" or "?".

5.2 Creating Files

Files are either automatically created by programs or they're created by users. In this section we'll present the MS-DOS commands that are used to create files.

Files automatically created by MS-DOS

Application programs usually store data in files. Some MS-DOS commands also create files. The following are some of the commands that create files:

SETUP	Installs MS-DOS and automatically creates the AUTOEXEC.BAT and CONFIG.SYS files.
BACKUP	Creates data and control files on backup diskettes. An optional parameter can be used to create a log file.
CHKDSK	Finds areas on a storage medium that aren't linked with files or are designated as unused, and converts these areas to files.
EDLIN	Makes backup copies of edited text files.
RECOVER	Places all legible files and file fragments in special files when trying to recover data.

Creating text files with MS-DOS

Text files may be created in MS-DOS with COPY CON, the supplied EDLIN program or the MS-DOS Editor program.

Use COPY CON to create text files

The COPY CON command copies keyboard entries to a file. To create a file, type the following command line:

```
COPY CON Filename
```

Then enter lines of text using the keyboard. End each line by pressing Enter. A line can contain up to 127 characters. To end the process and save the file, press either Ctrl + Z or F6 and Enter.

➡ If you enter a name that already exists, the new file overwrites the older file and the contents of the old file are lost.

Text files with EDLIN

Start EDLIN by entering the command and the name of the file to be created. An asterisk appears on the screen indicating that you're in the command mode of EDLIN. Enter I to move to input mode.

Now enter any text. Each line may contain up to 253 characters. Press Enter to move to the next line. To exit insert mode, press Ctrl + Z when the cursor is in an empty line. This will return you to the command mode. To save the file press E and Enter.

Creating files with EDIT

Enter EDIT to start the MS-DOS Editor. The MS-DOS Editor (new for DOS 5.0) will load. After pressing Esc to clear the welcome dialog box, you can create and edit text files.

Enter your text and press Enter to move to the next line.

To save the file, press Alt + F to choose the **File** menu. Move the cursor to the **Save as** command and press Enter. Enter your filename in the area that appears. Press Enter again to save the file.

To exit the editor, press Alt + F and then move the cursor to the **Exit** command.

For more information about working with the MS-DOS Editor, refer to Chapter 13.

Creating files by redirecting a command

There is another way to create files in MS-DOS. Ordinarily MS-DOS commands display information on the screen. However, the output of an MS-DOS command can be redirected to a file. You can use ">" to create a file for the output of a command.

Use the following command line to write all of the output of a command to a file instead of the screen:

```
Command >File
```

You can use any required parameters after the command. MS-DOS creates a file and redirects all screen output of the command to this file. Any existing file with the same name will be overwritten.

Certain commands cannot be redirected. Instead of displaying information to the standard output device (usually the screen), these commands use a special error output device.

Some programs write information directly to the video (screen) memory. The output from these programs also cannot be redirected to a file.

If you're developing programs with Turbo Pascal and want to arrange the screen output in such a way that it can be redirected to a file, you cannot use the CRT unit. Without this unit the screen display of the program uses the standard output device and can be redirected.

Example

Suppose that you want to print the contents of a directory and you also want to include some explanatory notes about the individual files. You could use the word processor EDIT or EDLIN to create a file containing a list of all the files in the directory.

However, it's much easier to create the file by redirecting the DIR command to a file and then editing the file.

To create this file, change to the desired directory (CD) and type the following command line:

```
DIR > CONTENTS.TXT
```

You can edit the file with either the MS-DOS Editor or EDLIN and then print it.

Be careful when redirecting commands to a file from a batch file. You can redirect direct commands, DIR > NUL. But you cannot redirect a batch file that handles certain commands. For example, you can't use DIRS.BAT > NUL to redirect a batch file named DIRS.BAT.

5.3 Copying Files

Copying files is important when working with your computer. When you copy files, the source files are preserved in their original location. For example, you could copy files from a hard drive onto a diskette, either for a backup or to copy the data onto another computer.

You can also copy files that only need to be changed slightly. If you only need to make a few changes to a document, it's easier to copy a file and edit it than to create a new file. This can be done with text files. For example, you can do this when you need to send the same letter to several people.

Various MS-DOS commands can be used to copy files. This chapter explains these options.

Copying files from a directory - COPY

The COPY command can be used to copy one or more files from a directory.

Copying files

To copy a file from the current directory to another directory, both the source file and the target directory must be specified.

```
COPY File1 Directory
```

This method of copying produces a copy of the source file in the specified directory. The copy has the same name as the original. After the computer is finished copying the file, MS-DOS indicates how many files were copied and the ready prompt reappears on the screen.

To copy a file from a directory other than the current directory, the entire path and the filename must be specified.

If a target directory isn't specified, MS-DOS uses the current directory as its target. If the file to be copied is also in the current directory and a new name for the new file hasn't been specified, MS-DOS reports an error because a file cannot be copied onto itself. In place of a target directory, specify a new filename.

Examples

If you want to copy the file INVOICE.TXT from the current directory onto a diskette in drive A:, type the following:

```
COPY INVOICE.TXT A:
```

Suppose that you want to copy the MS-DOS FORMAT command from the C:\DOS directory into the current directory. (The DOS directory cannot be the current directory; otherwise you'll receive an error message, since you would try to copy the file onto itself.) Since you want to copy to the current directory, you don't have to specify the name of the target directory:

```
COPY C:\DOS\FORMAT.COM
```

Use the following command to copy all MS-DOS commands, with the COM extension, from the DOS directory to the directory C:\PROGRAMS:

```
COPY C:\DOS\*.COM C:\PROGRAMS
```

☞ If the target directory doesn't exist, MS-DOS will display a corresponding error message.

Example

Suppose that you want to copy the file LETTER1.TXT from the current directory to the directory A:\DATA. However, the directory DATA doesn't exist. MS-DOS copies the file LETTER1.TXT to the main directory of drive A: under the filename DATA.

```
COPY LETTER1.TXT A:\DATA
```

Copying more than one file

Wildcards can be used to specify the filename for the source. MS-DOS will copy all files with matching filenames.

For example, copy all text files from the current directory onto a diskette in drive A: to provide a backup of your files. Do this by typing the following command:

```
COPY *.TXT A:
```

You might run into a problem if you wanted to create backups of files in the current directory and keep them in the current directory. Suppose that you have the following files in the current directory:

```
COMP01.TXT
COMP01.BAK
COMP02.TXT
```

You don't remember which files were located in the directory and now you want to back up all the files beginning with the letters COMP, use the following command:

```
COPY COMP*.* COMP*.BAK
```

The following message appears on the screen:

```
COMP01.TXT
COMP01.BAK
File cannot be copied onto itself
        1 File(s) copied
```

MS-DOS copied COMP01.TXT correctly, then tried to copy COMP01.BAK to COMP01.BAK. Since this is impossible, MS-DOS aborts the process with an error message and indicates that it could copy only one file (COMP01.TXT). To avoid this mistake, use the following (correct) command line:

```
COPY COMP*.TXT COMP*.BAK
```

☞ Using the COPY command can be dangerous because it overwrites an existing file in the target without asking you to confirm it. So use DIR to check whether there are any existing files in the target. This problem can be avoided by using the XCOPY command with the /P switch. You can also prevent files from being overwritten by copying files in the MS-DOS Shell.

Copying and checking files

The /V switch causes MS-DOS to verify the copies, which means that it tests whether the original file is identical to the copied file. Verifying files helps you determine whether any errors occurred during the copying process. The command line for verifying the copy operation looks similar to the following:

```
COPY File1 Directory /V
```

Using the verify option will ensure that the original file and the copied file are identical and will protect against any errors during a copy operation.

Copying files and changing filenames

By specifying a filename as the target, you can give files new names when they're copied. This is also the only sensible way to copy files within a directory:

```
COPY File1 File2
```

Wildcards can be used for both the source and the target. MS-DOS copies all files with the appropriate filename and changes the filename according to the default in File2. Here are a few examples:

Make a copy of all files with names beginning with the letter A, but change the first letter (A) of the filename to the letter B in the copy. Use the following command line:

```
COPY A*.* B*.*
```

Copy all files with the .TXT extension from the current directory to the main directory of drive A: and change the filename extension to .BAK by using the following command line:

```
COPY *.TXT A:\*.BAK
```

Select only those files you want to copy

When using wildcards, unwanted files might be copied accidentally. This problem can be solved by doing the following:

- Before copying, use the `DIR` command with exactly the same filenames as the ones to be copied. This ensures that you know which files will be affected by the copying process.

ASCII and binary file options

Use the /B and /A switches to make changes on files while copying them. The switches remain valid until another switch cancels them:

These two switches determine how end of file marks (Ctrl + Z = ASCII 26) are treated during the copy process. The end of file marks (EOF (End of File)) are used to determine when the end of a file has been reached by commands for pure text files (ASCII files). For programs or specially structured files from application programs, MS-DOS doesn't determine the end of the file by the end of file character, but determines the file size from the directory.

`/A`	After the source file: file up to end of file mark (Ctrl + Z = ASCII 26) is copied; data following the end of file mark is ignored.
`/A`	After the target file: adds an end of file mark to the end of the target file.
`/B`	After the source file: end of file mark is ignored; command copies file including all end of file marks.
`/B`	After the target file: no end of file mark at the end of the file.

The abbreviation /A represents ASCII and /B represents binary. When you use /A, MS-DOS treats the files like ASCII files that always end with Ctrl + Z. Since binary files can contain any characters, the character with the ASCII code 26 cannot signal the end of file.

Appending files

Another option when copying files is to combine several files into one file. This can be accomplished by specifying several files instead of a single source file. Separate the filenames with an addition symbol (+).

```
COPY File1+File2+File3 Total_file
```

Several files can also be combined into one file by using wildcards for the source and specifying a single filename as the target file. MS-DOS copies all files with the appropriate filenames to the new file.

If a name for the target file isn't specified, MS-DOS uses the name of the first source file. The original contents of this file are lost. MS-DOS displays a message on the screen informing you of this.

The /A and /B switches can be used when copying several files into one file. These switches produce the effects described above.

Copying files from more than one directory - XCOPY

The XCOPY command is used to copy files from a directory and its subdirectories.

This command is ideal for backing up a directory and its subdirectories onto a diskette. The advantage, compared to BACKUP, is that the files on the copied diskette are directly addressable from DOS.

The disadvantage of using XCOPY is that the number of files you can copy depends on the available space on the target storage medium.

Copying files

To copy the files of a directory with all of its subdirectories, use the following command:

```
XCOPY Source Target /S
```

The /S switch ensures that the files of the subdirectories are also copied. The structure of the directory is preserved when this method of copying is used. Files from subdirectories are copied into the correct subdirectory. If necessary, the directories are automatically created. When this method of copying is used, only the subdirectories that contain files are copied.

If the /S switch isn't specified, the command only affects files from the current directory.

The following paragraphs provide a more thorough explanation of the terms source and target when the /S switch is specified:

Source	If a filename is specified as the source, XCOPY copies this file from the current directory as well as any files with this name from the existing subdirectories. Wildcards can be used. If this option of XCOPY isn't wanted, specify the file by using the exact pathname. If a disk drive or a directory is specified as the source, XCOPY copies all files as well as the subdirectories and their contents.
Target	If a disk drive or a directory is specified as the target, the files are copied to that directory. If the source contains files in subdirectories, the subdirectories are also used in the target and, if necessary, MS-DOS creates them. If a target isn't specified, XCOPY uses the current directory. If a filename is used for the target, MS-DOS renames the files as it copies. Wildcards can be used.

If wildcards are used to select more than one file as the source, only one name is specified as the target. If there aren't any subdirectories with this name, you will be prompted to make the target unit either a directory or a file. If you specify the target unit as a directory, it will be created as a subdirectory of the current directory. If you specify the target unit as a file, only the last of the source files will be copied to the new file.

The following are a few more examples of working with XCOPY.

The following command line copies all files with the .TXT extension from the hard drive to a target diskette in drive A:. The command also sets up all of the necessary subdirectories on the target diskette.

```
XCOPY C:\*.TXT A:\ /S
```

The next command line copies the entire contents of the diskette in drive A: including any subdirectories to the DISKETTE directory of the hard drive:

```
XCOPY A:\ C:\DISKETTE /S
```

XCOPY Switches

Different switches can be specified to control the way XCOPY works. For example, switches can be set to select only certain files to be copied or even to control the copying process itself.

/A	Copies only files that have been created or changed since the last copying process (archive flag is set). The archive flag doesn't change.
/D:MM-DD-YY	Copies only files that have been changed since the specified date.

/E	Creates subdirectories in the target even though they might be empty.
/M	Copies only files that have changed or have been created since the last copying process (archive flag is set). The archive flag is reset.
/P	Gives a prompt for each file to be copied. Enter Y to copy.
/S	This switch must be set to copy subdirectories.
/V	Verifies the files in the target after the copying process.
/W	Requires a key press before the copy process will begin. This allows you to start XCOPY from the MS-DOS diskette and insert another diskette as your source.

Example

To copy all of the files and subdirectories, from the DATA directory, that have been created or changed since the last backup, to a diskette and designate them as saved by clearing the archive bit, use the following command line:

```
XCOPY C:\DATA\*.* A: /S /M
```

5.4 Comparing Files

MS-DOS provides two commands for comparing files: COMP and FC. The COMP command only compares files of the same length while the FC command compares files of any length. The number of lines, in the files to compare, can be specified with COMP by setting the /N switch. It may then be used to compare files of different length.

The results of the comparison operation may be displayed in hexadecimal format or as characters by setting the /D (display in hexadecimal format) or /A (display as ASCII characters) switches.

In either case, information about any existing differences will be provided. The default setting (no parameter set) displays the differences as hexadecimal.

Comparing files of the same size

Use COMP to compare two files of equal length. The following is the command line for this comparison:

```
COMP File1 File2
```

Path specifications and wildcards can be used for the names of File1 and File2. If only a path specification is entered for File2, MS-DOS will search this directory for a file with the same name as File1.

If File1 and/or File2 aren't specified (simply enter COMP), the command will ask for the required information with a series of prompts.

If COMP finds any differences, they are displayed in bytes. A message, similar to the following, appears:

```
Compare error at OFFSET 0
file1 = 48
file2 = 68
```

OFFSET displays the offset to the mismatched bytes. Next, the differing bytes in File1 and File2 are displayed as hexadecimal values.

In the previous example, the first mismatch was found at the first character in the file (OFFSET=0); File1 has a hexadecimal value of 48 and File2 has a hexadecimal value of 68 at this location.

The /L switch is used to display the line number of the different file instead of the offset to the mismatched file.

When comparing with COMP, there is usually a difference between upper and lowercase letters. When the /C switch is specified, the differences between upper and lowercase letters are ignored.

After ten mismatches, COMP stops the comparison for the selected files and displays the following message:

```
10 Mismatches - ending compare
```

After all of the specified files have been compared, answer the prompt by pressing Y to select more files for comparison.

If differences between the files aren't found, the command displays the following message:

```
Files compare OK

Compare more files (Y/N)?
```

When earlier versions of COMP were used to compare programs (binary files) instead of text files, the following message was displayed:

```
EOF Mark not found
```

EOF is an abbreviation for End Of File. MS-DOS uses the character with ASCII code 26 (Ctrl + Z) as an end of file mark for text files. Unlike text files, the size of binary files must be obtained from the directory. DOS 5.0 COMP does not display this message.

After completing a comparison, the following prompt appears on the screen:

```
Compare more files (Y/N)?
```

Answer Y; you're prompted to enter the names of additional files to compare.

Comparing files of different sizes

Use the FC command to compare files that are different sizes. Set the /B switch so that a binary comparison is performed. FC compares the files byte by byte. The command line is as follows:

```
FC File1 File2 /B
```

FC then displays the differences in hexadecimal notation:

```
Address Value1 Value2
```

It's best to compare only files whose contents are almost identical. As soon as a different character is found in a pair of files, error messages will be displayed. Here's an example of what can happen:

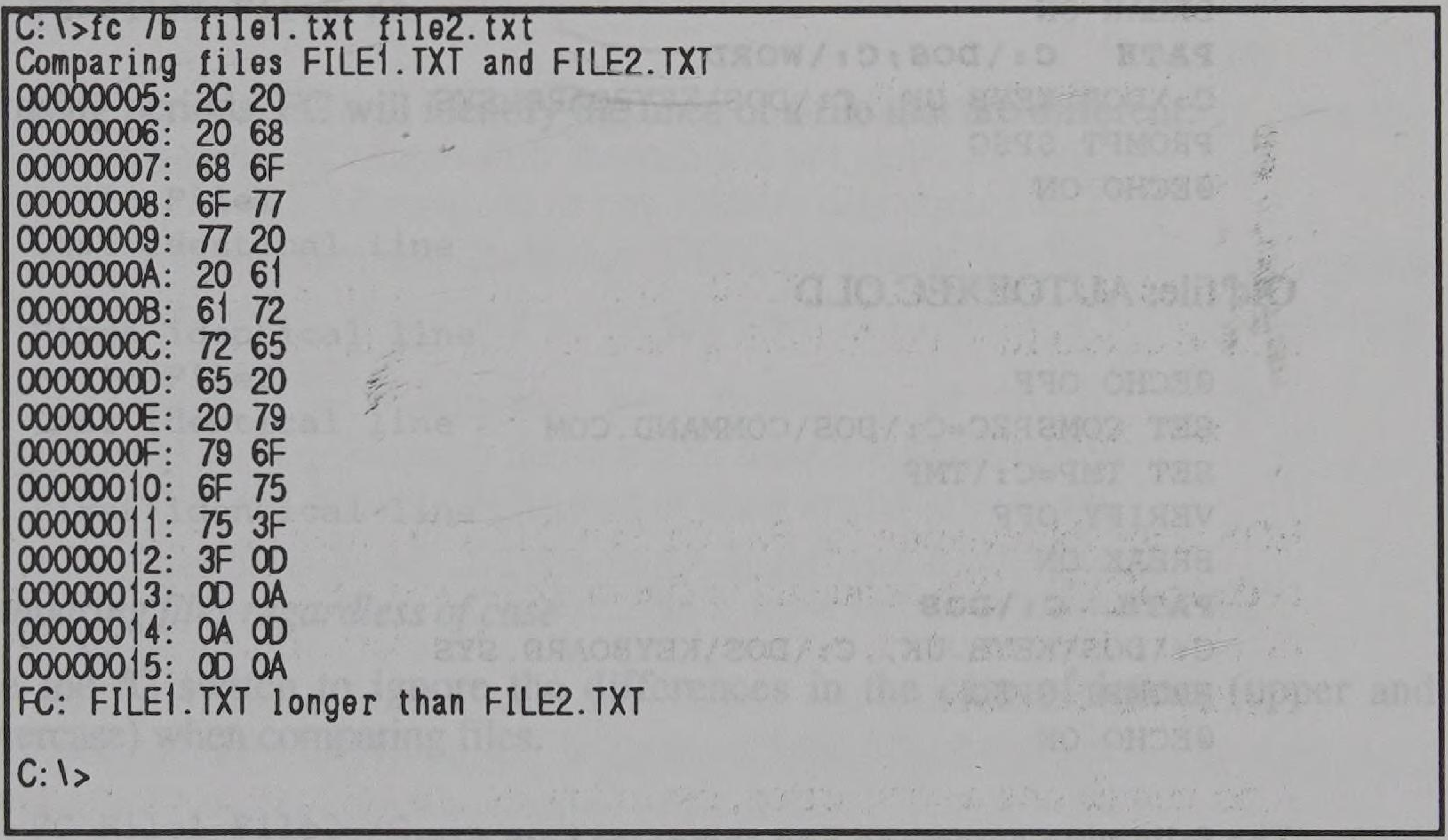
File1 contains the text:

```
HELLO, how are you?
```

File2 contains the text:

```
HELLO how are you?
```

FC displays the following messages:



```
C:\>fc /b file1.txt file2.txt
Comparing files FILE1.TXT and FILE2.TXT
00000005: 2C 20
00000006: 20 68
00000007: 68 6F
00000008: 6F 77
00000009: 77 20
0000000A: 20 61
0000000B: 61 72
0000000C: 72 65
0000000D: 65 20
0000000E: 20 79
0000000F: 79 6F
00000010: 6F 75
00000011: 75 3F
00000012: 3F 0D
00000013: 0D 0A
00000014: 0A 0D
00000015: 0D 0A
FC: FILE1.TXT longer than FILE2.TXT

C:\>
```

The only difference in the two files is that File2 is missing a comma. FC begins reporting differences at this point.

COMP can also be used to compare files of varying lengths. Specify the /N switch followed by a equal sign and the number of the lines to be compared. The COMP command displays the differences and the line numbers in which the differences occur.

Comparing two files

The syntax for using the FC command is:

```
FC File1 File2
```

Path specifications can be set for the files but wildcards cannot be used for specifying filenames. Only two files can be compared at a time.

Example

When Microsoft Word is installed, the installation program indicates that AUTOEXEC.BAT has been changed and the original file has been renamed to AUTOEXEC.OLD. To ensure that important lines aren't omitted, check both files for differences and changes. Although you can use an editor program to examine both files, FC is faster and easier.

New file: AUTOEXEC.BAT

```
@ECHO OFF
SET COMSPEC=C:\DOS\COMMAND.COM
SET TMP=C:\TMP
VERIFY OFF
BREAK ON
PATH  C:\DOS;C:\WORD
C:\DOS\KEYB UK,,C:\DOS\KEYBOARD.SYS
PROMPT $P$G
@ECHO ON
```

Old file: AUTOEXEC.OLD

```
@ECHO OFF
SET COMSPEC=C:\DOS\COMMAND.COM
SET TMP=C:\TMP
VERIFY OFF
BREAK ON
PATH  C:\DOS
C:\DOS\KEYB UK,,C:\DOS\KEYBOARD.SYS
PROMPT $P$G
@ECHO ON
```

The following command line compares both files:

```
FC AUTOEXEC.BAT AUTOEXEC.OLD
```

FC displays the section that has been changed:

```
***** AUTOEXEC.BAT
BREAK ON
PATH C:\DOS;C:\WORD
C:\DOS\KEYB UK,,C:\DOS\KEYBOARD.SYS
***** AUTOEXEC.OLD
BREAK ON
PATH C:\DOS
C:\DOS\KEYB UK,,C:\DOS\KEYBOARD.SYS
*****
```

If the specified files are different, FC tries to synchronize them. This means that it tries to find identical lines.

FC has a 100 line buffer for this purpose. If there are more than 100 lines that are different, FC ends the comparison and displays a message on the screen.

Even though there are a lot of differences between the files, you can continue the comparison by specifying the size of the line buffer. Use the /LB switch (line buffer) and specify the number of lines after the parameter.

At least two lines must match in order for FC to recognize the following partition of the files as identical. Use the /X switch to set the value which specifies the number of identical lines for a match.

To compare files without displaying the longer sections of the files that are different, use the /A switch:

```
FC File1 File2 /A
```

By using periods, FC will identify the lines of a file that are different:

```
***** File1
Last identical line
...
First identical line
***** File2
Last identical line
...
First identical line
```

Comparing files regardless of case

Use the /C switch to ignore the differences in the case of letters (upper and lowercase) when comparing files.

```
FC File1 File2 /C
```

FC converts lowercase letters into uppercase letters.

Comparison displaying the line numbers

The /N switch displays line numbers for differences found in files.

```
FC File1 File2 /N
```

For example, if the fifth line is different, the screen displays the following:

```
***** File1
     4:  Last identical line
     5:  Different line
     6:  First identical line
***** A:File2
     4:  Last identical line
     5:  Different line
     6:  First identical line
```

Special switches

The following is a table of all the switches that can be used with the file comparison commands. The switches for the COMP command are:

/A	Display differences as ASCII characters. This is only useful for comparing files that consist of printable characters.
/D	Display differences as ASCII codes in the decimal numerical system. A table of the ASCII codes for all characters is included in Appendix C.
/L	Specifies the line numbers containing differences. Instead of the line number, the command specifies the position of the character in the entire file.
/N:XXX	Determines that the command only compares the first XXX lines of both files. In files, the location of differences is indicated by line numbers. This switch also enables you to compare files of different length. Without using this switch, COMP will abort immediately when files are different lengths.
/C	If /C is specified, the command will ignore differences in uppercase/lowercase letters when comparing files. Usually differences in the case of letters leads to comparison errors.

The following switches are available for FC:

/A	Abbreviates the output, the first line is displayed, then the different regions in the files are marked "..." and finally the last line is output.
/B	Forces a binary comparison.
/C	Comparison regardless of upper/lowercase letters.
/L	Compares files as ASCII text.
/Lbxx	Sets maximum consecutive mismatches in xx lines.
/N	Displays line numbers during ASCII comparison.
/T	Tab characters aren't converted to spaces. Usually FC expands tab characters by spaces to the next tab stop.
/W	Compresses tab characters and spaces.
/nn	Sets the number of identical lines that must match after a mismatch. In order for the files to be considered identical, FC must find nn identical lines in both files after the mismatched area.

5.5 Viewing File Contents

Although ASCII files can be displayed on the screen, binary files will only display a small part of the file before special characters appear along with warning beeps. Even files prepared by programs, such as databases and word processors, contain characters which will create problems if displayed on the screen. Files can be viewed more easily by using the MS-DOS Shell, which will be explained later in this chapter.

Viewing a file

The TYPE command can be used to display files on the screen:

```
TYPE Filename
```

To pause a screen display, press Ctrl + S and to continue the display press the Spacebar or Ctrl Q.

The MORE filter can be used with the TYPE command by using the pipe symbol (|). The screen pauses after every screen page and then continues with the next page after any key is pressed.

Viewing one or more files

To display one or more ASCII files on the screen, use the following:

```
COPY File CON
```

Similar to the COPY command, wildcards can be used for filenames. MS-DOS displays the filename at the beginning of each file.

To display all of the batch files (files with the .BAT extension) in the current directory, on the screen, use the following command line:

```
COPY *.BAT CON
```

If more than one page is displayed, you can stop the display by pressing Ctrl + S. To set a page by page display, use the following command line:

```
COPY *.BAT CON | MORE
```

Viewing files with the MS-DOS Shell

The **View File Contents** command in the **File** menu can be used from the MS-DOS Shell to examine the contents of a file. If text files are displayed, the text is displayed in ASCII mode. MS-DOS Shell automatically uses HEX mode for program files and other non-ASCII text files.

By selecting options from the **Display** menu, files can be displayed in either HEX or ASCII mode.

5.6 Printing Files

Files can be printed directly or placed in a print queue. Using a print queue enables you to continue working with your computer while files are printed in the background.

☞ Files can be printed with the MS-DOS Shell by marking the files to be printed and selecting the **Print** command from the **File** menu. However, before this can be done, the printer must be initialized with the PRINT /D switch.

Printing files directly

To print one or more files directly, use COPY to copy the files to the printer port:

```
COPY File LPT1
```

Substitute the proper printer port if your printer isn't connected to LPT1. If necessary, configure the port. The filenames for the files that will be printed can also contain the path specifications. Wildcards can also be used to select more than one file.

Setting up a print queue for the printer

With PRINT, the print queue can be set up for the printing process. Although this allows you to continue working in MS-DOS, using a print queue will also slow down your PC slightly.

The print queue is defined the first time PRINT is called. While PRINT is using the print queue, the printer cannot be accessed with any other command.

Setting up the print queue

The first time PRINT is called, the print queue can be set up and the printer connection defined:

```
PRINT /D:LPT1
```

(Substitute the proper port if your printer isn't connected to LPT1. If necessary, configure the port as explained in the section on the Serial Port.)

If the /D switch isn't included, PRINT prompts you for the printer connection.

Defining the number of files and buffer size

Set the maximum number of files in the print queue and the size of the buffer the first time PRINT is called.

```
PRINT /D:LPT1 /Q:Number /B:Buffer
```

/Q:Number

If the number of files to be printed isn't specified, PRINT uses a default value of 10. The value for the number of files in the print queue can be between 1 and 32.

/B:Buffer

This switch allows you to set the size of the print buffer. The default value is 512 bytes. Generally, larger buffers allow the PRINT command to execute faster.

Determining waiting times

The first time PRINT is called you can define how much time is available for printing a character, how long PRINT waits for the printer and the amount of time PRINT can use for printing.

/M:Print time

Sets the number of time units available for printing a character. The default value is 2, but a value between 1 and 255 can be used.

/U:Waiting time

Determines how long PRINT waits before the printer can receive another character. The default value is 1, but a value between 1 and 255 can be used.

/S:ratio Sets the time ratio for the background process. The default value is 8, but a value between 1 and 255 can be used. When using the default value, there is eight times as much available time for working with the command interpreter as there is for the printing process. The higher the value, the more rapidly the PC reacts and the more slowly the printer prints.

You should experiment with the value for the ratio and set it so that you can work without being interrupted and the printing process will continue. Since printers vary in speed, there is no general value.

Changing the print queue

New files can be added and existing files can be deleted from the print queue. The list of files in the print queue can be displayed and the entire print queue can be deleted.

Adding a file to the print queue

To add a file to the print queue, simply specify the filename after the PRINT command:

```
PRINT Filename
```

Several files can be specified by separating each filename with spaces. Also, wildcards can be used for selecting and printing several files. This method can be used to add additional files to the print queue while PRINT is busy printing files.

Deleting files from the print queue

Files can be removed from the print queue by specifying the filename and the /C switch:

```
PRINT Filename /C
```

The /C switch affects the specified filename and all of the subsequent filenames in the print queue. For example, three files can be removed from the print queue as follows:

```
PRINT File1 /C File2 File3
```

☞ Using /C only removes the files from the print queue (from the print list); it doesn't delete them from the storage medium.

Adding files to the print queue and resuming printing

The /P switch adds the preceding filename and all subsequent filenames to the print queue:

```
PRINT Filename /P
```

The /P and /C switches can be combined in a single command line, which allows you to add and remove files simultaneously.

Example

The following command line removes the file WARNING.TXT from the print queue and adds the INVOICE.TXT file:

```
PRINT Warning.txt /C INVOICE.TXT /P
```

Deleting all files from the print queue

The /T switch is used to delete all files from the print queue and abort the printing process:

```
PRINT /T
```

5.7 Deleting Files

MS-DOS provides the DEL (Delete) and ERASE commands for deleting files. Since these commands are identical, we'll refer only to the DEL command in our explanations. However, remember that the ERASE command works the same way as the DEL command.

The DEL command

The DEL command can be used to delete one or more files from a storage medium.

```
DEL Filename
```

If a filename is specified, the command removes only the specified file. Path specifications can be used if the file isn't in the current directory.

Wildcards ("*" and "?") can be used to delete more than one file at the same time. However, if the /P switch isn't specified, the command deletes all of the files without displaying a security prompt.

Deleting all files in a directory

The wildcard *.* can be used to delete all of the files in either the current directory or a specified directory. In this case, the command will use a security prompt, which checks whether or not all of the files should be deleted. Enter Y for "Yes" to delete all of the files. Enter N for "No" to end the delete process.

The contents of an entire directory can be deleted by using the DEL command from a higher directory. For example, if you want to delete all of the files in the PERSONAL directory, which is a subdirectory of TEXTS, type the following if TEXTS is the current directory:

```
DEL PERSONAL
```

After a security prompt, the command deletes all of the files but leaves system, hidden and read-only files.

To delete the contents of any directory, specify the entire path. After a security prompt, the command will remove all of the files in the directory.

Special features for using wildcards

Always check which files will be affected by the deletion by using the DIR command. This is especially important when using wildcards because DEL will erase files without using additional security prompts.

For example, type:

```
DIR *.BAK
```

and then type:

```
DEL *.BAK
```

Prompt before deleting each file

The /P switch can be used to determine which files should be deleted. When this switch is specified, the screen displays each filename and a prompt before deleting the files. To delete the file press Y or to skip the file press N.

Undoing deletion of files

DEL doesn't remove the file from the storage medium. Instead, the first letter of the filename is replaced with the hex value E5. This marks the file as deleted and the space occupied by the file is made available for further use.

Recovering a single file

If no other files have been stored on the storage medium, the UNDELETE command can be used to recover the deleted file. To recover a deleted file, call UNDELETE with the filename of the deleted file. For example, the following command line would be used to recover an accidentally deleted TEST.TXT file:

```
UNDELETE TEST.TXT
```

The command will ask whether you really want to undo the deletion of the file. Answer the prompt with Y for "Yes". Then you're prompted to enter the first character of the filename. In our example this would be "T". MS-DOS then responds with a message indicating whether or not the file can be restored.

Recovering several files

Eventually you may accidentally delete several files in a directory. You can undo the deletion by calling UNDELETE, in the directory which contained the deleted files, and answering a prompt for each file. To recover a file, enter the first letter of the filename. After finishing this process, MS-DOS indicates whether or not the file was recovered.

☞ Always retrieve accidentally deleted files immediately. If any other files are saved, MS-DOS may overwrite the file(s) that were deleted. Once a file has been overwritten, it can no longer be recovered.

5.8 Renaming Files

Files within a directory can be renamed with the REN (RENAME) command. After a file is renamed, it's no longer available under its old name:

```
REN Oldfile Newfile
```

Oldfile	This parameter is the current filename. The name can include a path specification and a disk drive label. Wildcards ("*" and "?") can also be used.
Newfile	This parameter is the new filename. It cannot contain any path specifications. In other words, it must remain in the same directory. Any characters of the filename defined by wildcards remain unchanged.

If a file with the new filename already exists in the current directory, MS-DOS displays an error message:

```
Duplicate file name or file not found
```

Either select a new name for the file or use COPY.

Example

The following command line renames the file AUTOEXEC.OLD to AUTOEXEC.BAT. In order for the command to be successful, there shouldn't be an AUTOEXEC.BAT in the directory:

```
REN AUTOEXEC.OLD AUTOEXEC.BAT
```

To rename all files beginning with the characters LETTER so that the new filenames begin with COMP, use the following command line:

```
REN LETTER*.* COMP*.*
```

5.9 Special Features Of Files

MS-DOS stores special information (attributes) about every file. The ATTRIB command can be used to display and change this information.

Changing the write protect flag

The write protect flag determines whether or not a file can be changed or deleted. Use the ATTRIB command to set or delete this file attribute.

Protecting a file from being changed

To protect a file from being deleted or changed, use the command:

```
ATTRIB +R Filename
```

This file can no longer be deleted with DEL. Also, this file can't be specified as a target for the COPY command (it can't be overwritten by COPY). If you try to delete or copy to the file, MS-DOS displays the error message, "Access denied".

When specifying the filename, wildcards can be used to set the write protect flag.

Clearing the write protect flag

Use the following command to delete or change a file:

```
ATTRIB -R Filename
```

Setting and clearing the write protect flag in subdirectories

The /S switch can be set to process all of the specified files and associated subdirectories in the current or specified directory:

```
ATTRIB +R Filename /S
```

If the filename doesn't have a path specification, the command affects the current directory and any subdirectories.

Changing the archive flag

ATTRIB can also be used to set or clear the archive flag. A set flag means that the specified file has been changed or created since the last backup. If a file has this attribute set, you should create a backup copy as soon as possible. MS-DOS changes the archive flag automatically.

The archive flag can also be used for selecting files to use with the XCOPY, BACKUP and RESTORE commands.

Marking a file as changed

To mark a file as changed, use:

```
ATTRIB +A Filename
```

Wildcards can be used in the filename. Using wildcards enables you to change the attributes of several files simultaneously.

The command sets the archive flag of a file. The file still hasn't been changed; it is only marked as changed.

MS-DOS automatically sets the archive flag whenever files are created or changed. Sometimes it's helpful to set the flag later. For example, you may want to back up (even if it wasn't changed) or edit the file.

Marking a file as backed up

To mark a file as backed up, use the following command:

```
ATTRIB -A Filename
```

The command clears the archive flag of the file. This doesn't back up the file. Instead, the file only appears to have been backed up with the XCOPY, BACKUP and RESTORE commands (if you use the appropriate parameters with the commands). These commands may be configured to ignore files that have the archive flag cleared.

When changes are made to a file, MS-DOS automatically sets the archive flag.

If you've changed files and don’t want to back them up, clear the archive bit for these files before performing a backup.

Example

Several files, which are located in the C:\DOC directory, have been edited. Make a backup copy of these files to create a special print file called PRINT.TXT. The PRINT.TXT file doesn't need to be backed up. The following command line can be used to exclude PRINT.TXT from the backup:

```
ATTRIB -A PRINT.TXT
```

Now the remainder of the changed files can be backed up by typing the following command line:

```
XCOPY *.TXT A: /M
```

The /M switch used with XCOPY, determines that only files with set archive flags should be backed up. The command marks the files as backed up (XCOPY automatically resets the archive flag).

Changing the archive flag for files in several directories

The /S switch allows you to change the attributes of the specified files in the current or specified directory and any subdirectories:

```
ATTRIB +A Filename /S
```

This command line sets the archive flag for all of the corresponding files, including files in any subdirectories.

```
ATTRIB -A Filename /S
```

The command line above clears the archive flag for all corresponding files, including files in any subdirectories.

If the filename doesn't include a path specification, the command affects the current directory and any subdirectories. If a path specification is included, the command affects the directory specified in the command line.

Changing the hidden attribute

The hidden attribute marks a file as hidden. This means that the DIR command no longer displays the filename and the file cannot be removed by using DEL. However, there is a way to display files with this hidden attribute set. Call DIR as shown in the following command line:

```
DIR /A:H
```

Setting the hidden attribute

To set a file as hidden, use the following command:

```
ATTRIB +H Filename
```

Wildcards can be used in the filename.

Clearing the hidden attribute

The following command is used to clear the hidden file attribute and allows you to handle the file as any normal file.

```
ATTRIB -H Filename
```

This command clears the hidden attribute, allowing you to use the DIR and DEL commands.

Changing the hidden attribute for files in several directories

By using the /S switch, specified files in the current directory and any subdirectories can be affected.

```
ATTRIB +H Filename /S
```

The command line shown above sets the hidden attribute for all corresponding files, including files in any subdirectories.

```
ATTRIB -H Filename /S
```

This command line deletes the hidden attribute for all corresponding files in the current directory and any files in subdirectories.

If the filename specification doesn't include a path specification, the command affects the current directory and any associated subdirectories. If a path specification is included, the command affects only the specified directory.

Changing the system attribute

The system attribute marks files as system files. Certain programs treat system files differently. For example, the BeckerTools Compress program for optimizing hard drives considers system files as unmovable.

Setting the system attribute

To mark a file as a system file, use the command:

```
ATTRIB +S Filename
```

Wildcards can be used in filenames.

Clearing the system attribute

The following command line clears the system attribute, which allows the file to be handled as a normal file.

```
ATTRIB -S Filename
```

Changing the system attribute for files in several directories

The /S switch allows you to edit all of the specified files in a directory and its associated subdirectories:

```
ATTRIB +S Filename /S
```

The above command line sets the system attribute for all files in the current directory and any files in any associated subdirectories.

```
ATTRIB -S Filename /S
```

This command line clears the system attribute for all files in the current directory and any files in any associated subdirectories.

If the specified filename doesn't include a path specification, the command affects the current directory and all of its subdirectories. Otherwise the command affects only the specified directory.

Displaying file attributes

To display the file attributes of one or more files, use the ATTRIB command with the filename only, as shown below:

```
ATTRIB Filename
```

Wildcards can also be used in filenames. Use the /S switch in the command line to display the attributes of files in the subdirectories.

5.10 Checking Files

In this section we'll discuss ways to check the integrity of files. When this is done, files are checked for readability and for optimum distribution on the storage medium.

Checking files for readability

You should occasionally check files for readability. For example, suppose that some files have been copied onto a diskette. When the DIR command is used, the filenames are displayed. However, when you try to access the files, an error message appears.

This could be either a nuisance or a fatal error, especially if you were planning to copy these files onto someone else's computer. So, you should check the files on the diskette for readability.

The COPY command will check a file for readability. If MS-DOS cannot read parts of the file correctly, it automatically displays an error message. The fastest, most effective way to check file integrity is by using the dummy device NUL, as shown in the following command line:

```
COPY File NUL
```

Wildcards can be used when specifying the filename.

The previous command line copies the file to "nothing". The copying process itself is executed. However, since the copying process is extremely fast, no files are actually created.

If MS-DOS is no longer able to read a file correctly, use the RECOVER command to recover the file. Additional information about this command will be provided in later sections.

Checking files for optimum distribution on the storage medium

If it ever seems like your hard drive is slowing down as time goes on, it probably is. Hard drives become slower as more data is placed on them. The more often data is deleted and new data files are created, the more apparent the problem will become. To understand this completely, you need to know a little about how MS-DOS stores data on a hard drive.

If the hard drive is empty, MS-DOS stores data contiguously. This means that MS-DOS writes data on a track of the hard drive until it is full, then starts storing data on the next track.

Since a hard drive usually has at least two disks with two read/write heads each for the top and the bottom, MS-DOS can write on four tracks before the read/write head must be moved. These tracks, which can be processed without moving the heads, are called "cylinders".

On a typical disk drive with 17 sectors per track you can fit 17 * 512 or 8,704 bytes on a track. So 34,816 bytes can be stored on the four tracks of a cylinder. Since the computer can read or write to a file with 34K without having to move the read/write heads, the process is very fast.

Also, it's ideal if files are distributed consecutively on the disk sectors. When this occurs, the heads don't have to be moved several times, which causes waiting times while the heads "seek" a new position.

File Fragmentation

Files are constantly being deleted and saved on the hard drive. Even within the same file changes can be made to the text and saved. When a file is deleted, empty regions develop on the hard drive. Later these regions are used again to store data. If the newly saved files are bigger than the amount of free, contiguous space, the file is scattered over different places on the hard drive.

When the computer reads or writes to these files, the head must be moved to new tracks continuously, so that it can read or write other parts of the file. This causes performance to degrade considerably.

This process can be difficult if a hard drive is almost full. In this case there are only several small areas in which MS-DOS can distribute the files. For this reason you should never fill your hard drive to more than 90% of its capacity.

Checking files for contiguity

Naturally, the less your files are fragmented over the storage medium, the faster MS-DOS can read them. MS-DOS reads files faster when they're saved contiguously.

To check the distribution of a file on a storage medium, use the following command line:

```
CHKDSK File
```

CHKDSK displays the usual information about the storage medium and the memory of your computer and indicates whether or not the file is contiguous:

```
All specified file(s) are contiguous
```

The previous message indicates that the specified file is optimally allocated.

```
File contains XX non-contiguous blocks
```

Instead of being allocated contiguously on the diskette, the specified file consists of XX fragments.

Wildcards can be used for filenames.

Combining files in DOS

If the storage medium contains many fragmented files, the BACKUP and RESTORE commands can be used to allocate them contiguously. Use the BACKUP command to make a security copy of your entire hard drive. After a backup copy has been made, delete all of the files on the hard drive and use RESTORE to copy the files`back. MS-DOS automatically allocates the files contiguously. The only problem with this process is that it's time-consuming. We recommend using special hard drive optimizing programs such as those included with BeckerTools or Norton Utilities.

5.11 Searching For Files

File searches can be performed with either DIR or CHKDSK.

Searching for files in a directory

To search for files in a directory, enter the filename along with DIR. If the file isn't in the current directory, specify the drive and directory as well.

Searching for files on the entire storage medium

Use DIR from the root directory to search for files on the entire storage medium. After the command enter the name of the file you're searching for and the optional /S switch. The /S switch specifies that the command will search for all subdirectories.

```
DIR FILENAME /S
```

5.12 Searching Contents Of Files

The FIND command can be used to search one or more files for a specified text string.

Displaying lines containing a specified search text

Use the following command line to display all of the lines of a file that contain a certain search text :

```
FIND "Search text" File
```

FIND displays the name of the file and all of the lines in the file containing the search text:

Example

To search the AUTOEXEC.BAT for the term ECHO, use the following line:

```
FIND "ECHO" AUTOEXEC.BAT
```

Depending on the contents of your AUTOEXEC.BAT file, the following information is displayed:

```
---------- AUTOEXEC.BAT
@ECHO OFF
@ECHO ON
```

The search text must be enclosed in quotation marks. If the character sequence for the search is already enclosed in quotation marks, a double set of quotation marks must be used. The FIND command is case sensitive, which means that it is affected by upper and lowercase letters. If the /I switch is specified, the case of the letters (uppercase or lowercase) will be ignored.

To search more than one file, enter each filename separated by a space:

```
FIND "Search text" File1 File2 File3
```

Special options for searching

By specifying optional switches it's possible to utilize other features:

`/C`	Displays only the number of lines containing the search text.
`/N`	Displays the line number in front of each line containing the search text.
`/V`	Reverses the search condition. Displays only lines that don't contain the search text.
`/I`	Ignores the case of letters (upper and lowercase).

5.13 Working Faster With Files

Before you can work with a file, it must be opened. This process can be time-consuming, especially if your computer has extensive directory structures. To speed up this process, use FASTOPEN.

FASTOPEN decreases the amount of time needed to open frequently-used files and directories. This command keeps track of the location of files and directories on the storage medium and caches the information in memory for quick access. If the file is re-opened, the information stored by FASTOPEN is used. This significantly reduces access time.

To activate FASTOPEN, use the following command line:

```
FASTOPEN Disk_drive
```

FASTOPEN can be used with up to 24 hard drive partitions at one time and can track up to 999 files. If more files are opened than have been specified by FASTOPEN, MS-DOS removes the files that haven't been used for the longest time and adds new files. Information about each file that is opened and being tracked by FASTOPEN uses about 48 bytes of memory.

To use FASTOPEN on drive C: (the hard drive), use the following command line:

```
FASTOPEN C:
```

FASTOPEN can be called only once per work session. To change settings, restart your computer.

☞ FASTOPEN only works on hard drives.

The number of files that FASTOPEN tracks can be specified when calling the command:

```
FASTOPEN Disk_drive=Number
```

Use values between 10 and 999. Since FASTOPEN requires approximately 48 bytes of memory for each file, don't use values that are too large because you'll reduce the amount of available memory for your application program.

To increase the access speed for up to 100 different files on drive C: use the following command line:

```
FASTOPEN C:=100
```

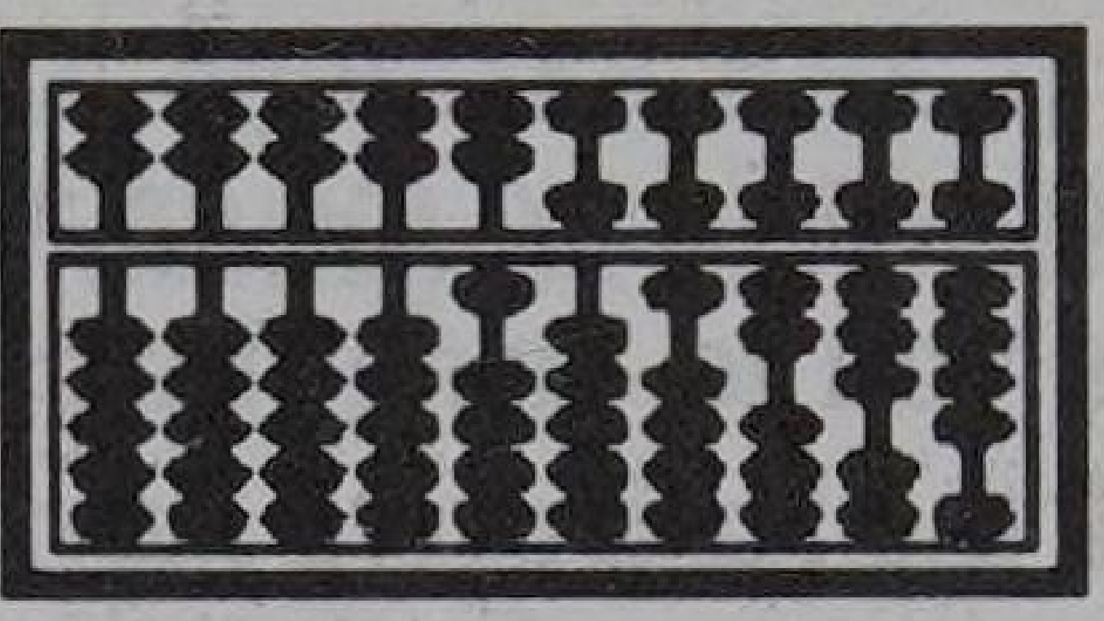
➡ FASTOPEN memorizes where the files are located on the storage medium. You may have problems if you use utility programs to reorganize and optimize your hard drive. FASTOPEN shouldn't be used with these utility programs.

When FASTOPEN is called with the optional /X switch, the filename cache is created in the expanded memory. This leaves more main memory available for application programs.

With the following line in your CONFIG.SYS file, FASTOPEN can be installed automatically at the start of every session and will track 100 files on drive C:

```
INSTALL=C:\DOS\FASTOPEN.EXE C:=100
```

Chapter 6



Backing Up And Recovering Data

When everything on your computer is running smoothly, you probably don't think about backing up your work. However, once work that took days or weeks to complete is suddenly destroyed, you'll probably realize that it has been a while since you backed up your work. In this chapter we'll discuss the important topic of data security.

6.1 Why Backup Your Data?

Perhaps you didn't expect to find a chapter containing information on data security in this book. However, if several weeks or months of work is suddenly destroyed, it's comforting to know that you have backup copies. In this chapter we'll show you the safest, easiest ways to back up your valuable data.

Before we begin let's briefly discuss two misconceptions about data security.

Many people believe that the only way to destroy data is by accidentally entering DEL *.* or FORMAT C:. However, since the dangers of using these two commands are well known, the chances of someone accidentally deleting files or formatting a hard drive are relatively small.

Entering DEL *.* destroys only the files in the current or specified directory. The damage can be undone rather easily with special tools, including MS-DOS 5.0's UNDELETE command. Most hard drives and versions of DOS allow you to abort FORMAT C:, while the command is running, without experiencing any data loss. Also, in some cases, the UNFORMAT command can be used to recover from an accidental format.

Unfortunately, there are more dangerous threats to your data than these two commands. The following are a few hypothetical examples of how data could be lost.

1. Suddenly your hard drive begins to act strangely. There seems to be a problem with the hard drive controller. Data can be read, but it's impossible to write to the disk.

   Since you need the data urgently, you call your dealer for help. The dealer tells you to bring the computer to his store. In your haste, you forget to park the hard drive before moving it.

   When you arrive, the computer won't operate with the hard drive. Because the disk was subjected to vibration during the trip to your dealer, the read/write head has been damaged.

   Your irreplaceable system information and the information stored on the disk has been permanently destroyed.

2. While working with a word processing program, it suddenly crashes while saving a file. At first, this doesn't seem to be a problem because the program automatically creates BAK files (backup copies of the existing file). However, the next time you boot your computer, it begins to behave strangely and error messages are displayed on the screen.

   As a check, you run CHKDSK which informs you that there are over 2,000 lost areas on your disk. Actually, CHKDSK has found that most of the files on the hard drive are either empty or totally useless, including your word processor's BAK file. Unfortunately, you don't have any backup copies on diskettes.

Perhaps these examples will convince you that the data on your hard drive is about as safe as leaving your wallet in an unattended shopping cart at the supermarket.

Quick and easy backup

When the subject of creating hard drive backups is discussed, many users immediately think of inconvenient, tedious processes. However, in most cases there are methods for providing data security that are a lot faster than the usual method, which involves swapping several backup diskettes.

In the following section we'll discuss different ways you can prevent data loss.

6.2 Backing Up The Hard Drive

Backing up data is extremely important when you use a hard drive. Unlike diskettes, DISKCOPY can't be used to make a copy of the hard drive. However, MS-DOS does provide some special commands for backing up the data on your hard drive.

The BACKUP command can be used to copy the data on your hard drive to diskettes. If the data on your hard drive is destroyed, the RESTORE command can restore files from the diskettes.

BACKUP stores the files on diskettes in a special format that can only be used by RESTORE to recover the files. The files created by BACKUP are write protected and cannot be deleted directly.

In order to use BACKUP diskettes for other purposes, either use FORMAT to format the disks or ATTRIB to clear the files for deletion.

To back up all of the data from a hard drive (files, directories and their structure), use the following command line:

```
BACKUP Root_directory Backup_drive /S
```

Instead of "Root_directory" enter the disk drive label followed by a backslash and instead of "Backup_drive" enter the drive that will be used by the backup (A:).

/S	Also include any subdirectories because BACKUP automatically backs up the files in the subdirectories. This preserves the directory structure.

Beginning with MS-DOS 4.0, the BACKUP command automatically formats diskettes. In earlier versions of DOS (such as MS-DOS 3.3) you must specify the /F switch as an additional switch for the diskettes to be formatted automatically.

Be sure that you have enough blank diskettes for your backup. Use CHKDSK to display how much space the data takes up on the hard drive and then calculate how many diskettes are needed. If you are using 360K diskettes, base your estimate on 350K of data per diskette, instead of 360K. Since BACKUP places control information on the diskettes, the entire capacity of the diskette isn't available for storing data.

Example

To back up all files from drive C: onto diskettes in drive A:, use the following command line:

```
BACKUP C:\ A: /S
```

MS-DOS prompts you to place the backup diskettes in drive A: and press any key.

BACKUP deletes all of the files in the root directory of the backup diskettes. If you use the same backup diskettes for another backup, the previous contents of the diskette will be lost. Don't use BACKUP if you have joined or assigned drives or directories with the MS-DOS commands SUBST, ASSIGN and JOIN, because this could cause errors. Since these drive redirections could cause inaccurate backups of your data, they should be corrected before proceeding.

BACKUP only deletes the existing files in the root directory of the diskette. Subdirectories and their files on "target diskettes" remain intact, which decreases the storage capacity for the backup. If this is the case, BACKUP will require additional diskettes. You may want to reformat diskettes containing subdirectories before using them with the BACKUP command.

Remember to label and number the backup diskettes. If you don't do this you may have trouble recovering the files with RESTORE. The backup diskettes must be inserted in the same sequence that was used when the backup was created.

If your hard drive is divided into several partitions, a backup must be performed for each disk drive.

How often should you perform a complete backup?

The time required for a complete backup of a hard drive depends on the amount of data and the format of the diskette. Fortunately, it's not necessary to perform a complete backup very often.

Over a period of time you'll make fundamental changes to the data on your hard drive. For example, you will probably purchase new programs and install them on your hard drive or replace existing programs with newer versions. If several months have passed since the last time you performed a complete backup, you should back it up again.

Remember, you can always reinstall programs that were installed since the last backup. So losing data doesn't cause you to lose programs. However, it's time-consuming to install these programs. Also, most users set up programs according to their own requirements. So when reinstalling programs you must adapt and set up programs according to the requirements you previously used.

6.3 Restoring All Files

Data loss can occur for many reasons. However, if your data has been backed up, your hard drive can be restored quickly.

Use the RESTORE command to copy your backup diskettes, created by BACKUP, onto your hard drive. The files and directories will be the same as they were at the time the backup was performed. Any changes made since the last backup are lost (unless you saved those changes in some other way).

When files are restored, data can also be lost if the command overwrites existing files. So don't accidentally increase the amount of damage already done. Fortunately, an optional switch (/N) prevents files from overwriting existing data.

Don't use RESTORE if you have redirected or assigned drives or directories with the MS-DOS SUBST, ASSIGN or JOIN commands.

RESTORE places data in the same directory in which it was stored. The command creates exactly the same directory structure that existed when the backup was made. If the directories no longer exist, they are automatically created.

To restore all of the data lost on a backed up storage device without changing any of the existing data, use the following command line:

```
RESTORE Backup_drive Root_directory /S /N
```

In place of "Backup_drive", substitute the drive from which you want to copy back the files. Usually A: is used here.

Instead of "Root_directory", enter the disk drive label followed by a backslash.

/S	Restores files in subdirectories.
/N	Copies only files that don't exist in the target directory. This means that the command doesn't overwrite any files that are still on the hard drive.

Place the backup diskettes in the drive one after the other and press any key. On the screen RESTORE displays the names of all of the restored files.

Example

To copy the data from backup diskettes in drive A: back to drive C:, without overwriting existing files, use the following command line:

```
RESTORE A: C:\ /S /N
```

To restore all files, regardless of whether the files exist on the target drive, use the following command line:

```
RESTORE A: C:\ /S
```

6.4 Backing Up Selected Files

Since it isn't necessary to back up all of the data of a storage device, certain directories or files can be specified. This can significantly reduce the time needed to perform a backup. Although it's only necessary to perform a complete backup of the hard drive occasionally, you should frequently back up data that is always being changed.

Securing only files from one directory

To backup files from only one directory, omit the /S switch:

```
BACKUP File A:
```

It's also possible to specify paths and disk drive labels. Wildcards can also be used. If only the drive letter (E:) is specified, BACKUP only saves files from the current directory of drive E:.

Examples

This example backs up all files from the DOS directory of the hard drive (C:) onto diskettes in drive A:

```
BACKUP C:\DOS A:
```

The following example backs up all programs, with the .COM extension, from the DOS directory of the hard drive onto diskettes in drive A:

```
BACKUP C:\DOS\*.COM A:
```

In order to back up all files, with the .DOC extension, from the current directory of the hard drive onto diskettes in drive A:, use the following command line:

```
BACKUP C:*.DOC A:
```

Backing up specific files

To back up only files with a specific filename from the hard drive, use the following command line:

```
BACKUP Disk_drive\File A: /S
```

By specifying /S, BACKUP will search all of the subdirectories of the entire storage device for files of this name and copy them onto a diskette. Wildcards can be used with filenames.

If the backslash is omitted, BACKUP copies files from the current directory of the drive and all of its subdirectories. If the path specification is extended, BACKUP copies all of the appropriate files beginning from the path specified and including all subdirectories.

Examples

The following example copies all the files, with the .DOC extension, on the hard drive onto diskettes in drive A:

```
BACKUP C:\*.DOC A: /S
```

To copy all the files, with the .DBF extension, on the hard drive (C:) from the current directory and all of its subdirectories onto diskettes in drive A:, use the following command line:

```
BACKUP C:*.DBF A: /S
```

This last example backs up all filenames, beginning with BOOK and ending with the .DOC extension, from the C:\DOC directory and its subdirectories onto diskettes in drive A:

```
BACKUP C:\DOC BOOK*.DOC A: /S
```

Backing up changed files

The /M switch can be used with the BACKUP command to copy only new files or files that have been changed (modified) to the backup diskettes. The following example copies all of the files that have been changed on the hard drive, including subdirectories, onto backup diskettes in drive A:

```
BACKUP C:\ A: /M /S
```

BACKUP uses the MS-DOS archive attribute to determine whether the file has been changed. When BACKUP copies a file onto a diskette, it automatically clears the attributes of the file. If you create a file with a user program under MS-DOS, the archive attribute is automatically set.

It's not always necessary to back up the entire hard drive onto diskettes. Often it's sufficient to back up only those files that have been changed. These files can be copied to a diskette or, with some special limitations, the files can be copied to a special backup directory on the hard drive.

If your hard drive has a lot of empty space, this space can be used for quickly backing up your current data. You should create a target directory, called \BACKUP, for this. Backups can be made to this directory on a regular basis. The MS-DOS XCOPY command can be used to do this. XCOPY also copies the directory structure.

Example

To increase the usefulness of the following example, set up a directory (with subdirectories), on your hard drive, in which you can "preserve" all of the files that are processed by user programs.

Create a directory, called \DATA, that contains subdirectories named \WORD, \DBASE, \DOC, \GRAPHICS, etc. These subdirectories are the "DATA directories" of your user programs. This means that anything you create and change (edit) with these programs is located in a subdirectory of the \DATA directory. If you have this structure on your hard drive, simply type:

```
XCOPY C:\DATA\*.* C:\BACKUP /M /S
```

The SAVE.BAT file

To further simplify this process, place this command line in a batch file named SAVE.BAT. Whenever you want to back up your data quickly and easily, simply type:

```
SAVE
```

The /M and /S switches have the same meaning as when they're used with BACKUP. Specifying /M backs up all files that have the archive flag set and specifying /S causes all subdirectories (of \DATA) to be included. Using the command line with these switches helps you back up your current files quickly and easily.

This method can be improved by using an extra partition (made by FDISK) of the hard drive. For example, if you have a 40 Meg hard drive, you can partition 32 Meg as drive C: and the other 8 Meg as drive D:. Then copy the files of the \DATA directory from drive C: to the \BACKUP directory on logical drive D:. The command line is as follows:

```
XCOPY C:\DATA\*.* D:\BACKUP /M /S
```

This command line can also be located in the SAVE.BAT file.

Having a second partition as a backup drive is extremely useful. If important system information is destroyed on the actual data drive (in the File Allocation Table, the directory, etc.), data saved in a special backup directory can also be destroyed. However, if this backup directory is on a different partition with its own FAT and directory, the data will still be usable.

Backing up files from a specific date

To copy only files from a specific DATE, the optional /D: switch is used with BACKUP:

```
BACKUP C:\ A: /D:DATE /S
```

For example, to back up all files, that were created or changed after 9/3/1990, from drive C: onto backup diskettes, use the following command line:

```
BACKUP C:\ A: /D:09/03/90 /S
```

In order to use the date or time options, you must set the DATE and TIME correctly each time you start your computer if you don't have a battery operated clock.

Special switches of BACKUP

Along with the previously mentioned switches, BACKUP can be used to add files to existing backup diskettes and can record the files that are backed up.

/A	Adds files to existing backup diskettes. Instead of deleting the contents of the backup diskettes, this process adds new data since the last backup.
/L	Creates a log file of backed up files that can be viewed by using the TYPE command. A pathname can be specified for the log file. If a path isn't specified, the file is stored in the root directory of drive C:. Although the default file is called BACKUP.LOG, you can specify an alternative filename after the /L: switch.

You should copy the log file to the last backup diskette so that you can easily determine the contents of the backup diskettes.

The date and the time of the backup are placed at the beginning of the backup log file. All backed up files, with complete path specifications (no disk drive label), and the number of the backup diskette are displayed.

```
7-14-1990  12:55:14
001  \DOS\BACKUP.COM
001  \DOS\CHKDSK.COM
001  \DOS\COMP.COM
001  \DOS\DISKCOMP.COM
```

Instead of overwriting an existing control file when a new backup is performed, MS-DOS adds the data of the new backup to the existing control file. If you want to keep only the data of the current backup in the control file, delete this file before performing a backup.

/T:Time	Backs up only files created after the specified time. If a DATE is also specified, the /T switch is ignored.

In the section "Backing up changed files", we showed you how to back up data quickly by using the XCOPY command. The following batch file, D-BACKUP.BAT, performs this task and allows you to specify the date and another drive besides the hard drive.

```
@ECHO OFF
REM D-BACKUP.BAT
REM by Manfred Tornsdorf
REM Check number parameters
IF "%3" == "" GOTO Syntax
If NOT "%5" == "" GOTO Syntax
REM Note /S in UV variable
SET UV=%4
GOTO Start
REM Incorrect number parameters
:Syntax
ECHO D-BACKUP.BAT backs up changed files beginning at a
certain date
ECHO Syntax: D-BACKUP source target date [/S]
ECHO To include subdirectories add "/S"
GOTO End
:Start
XCOPY %1 %2 /D:%3 /M %UV%
ECHO Finished!
:End
REM Delete used variables
SET UV=
ECHO ON
```

Place the D-BACKUP.BAT batch file in a special batch directory or in the directory containing the MS-DOS commands.

Enter the directory and/or files after the D-BACKUP batch command as the source. Then specify the target directory or drive. The last parameter the batch file expects is the date [mm-dd-yy] that the command should use to begin backing up the files that have been changed.

MS-DOS remembers files that have been changed

After copying files, MS-DOS clears the archive attribute for each file copied. MS-DOS uses this bit to indicate whether changes were made to a file. The archive bit remains cleared until new changes are made to the file.

If you consecutively call D-BACKUP.BAT twice with the same specifications for the source, the command backs up the files the first time, but not the second time. This occurs because all the files have been backed up and none of the files have been changed.

If you add /S as the last switch, the command also affects subdirectories of either the specified directory or the current directory.

Examples of D-BACKUP.BAT

The following command line backs up all files, that have been changed (beginning with 5/3/90), onto a diskette. It doesn't back up older files or files that have already been backed up.

```
D-BACKUP *.* A: 05/03/90
```

The following line copies all changed DOC files, beginning with 5/3/90, from the C:\DATA directory and its subdirectories to the C:\BACKUP directory of the hard drive:

```
D-BACKUP C:\DATA\*.DOC C:\BACKUP 05/03/90 /S
```

To back up all changed files regardless of the DATE, simply use the earliest possible value for the DATE: 1/01/1980. If you want to use the D-BACKUP.BAT option to back up data to a diskette, remember that XCOPY, unlike BACKUP, can only use a single diskette.

6.5 Restoring Selected Files

You don't have to restore all of the data that has been backed up. It's possible to specify certain directories or files. Since RESTORE places the data back in the same directory from where it was backed up, the data can't be stored in a different directory.

Restoring files from only one directory

To restore only files from one directory, use the following command line:

```
RESTORE A: File
```

Enter the path and the disk drive label for "File". Wildcards are also allowed. If you only specify a drive for "File", the command restores only the files in the current directory.

Example

The following example restores all of the files, that begin with BOOK and have the extension .DOC, from the DOC directory of the hard drive C:. The command doesn't consider subdirectories.

```
RESTORE A: C:\DOC\BOOK*.DOC
```

Special switches for restoring files

Other switches, which enable you to select files according to different criteria, can be specified.

/P	If the file to be replaced is write protected or if you've made changes to it since the last backup, RESTORE indicates this and asks for confirmation.
/A:Date	Restores only those files that have been changed since the specified date.
/B:Date	Restores only those files that were changed before the specified date.
/L:Time	Restores only files that have been changed since the specified time.
/E:Time	Restores only files that were changed before the specified time.
/M	Restores only those files that were changed since the last backup.
/N	Restores only files that no longer exist on the target disk.

Example

Suppose that you accidentally deleted some files, with the .DOC extension, in the C:\DOC directory and want to restore them. However, you don't want to overwrite text files with older versions from the backup diskettes:

```
RESTORE A: C:\DOC\*.DOC /P
```

The command restores all text (.DOC) files that no longer exist. There is a security prompt for files that have been changed. Press N to avoid copying changed files.

6.6 Automatic Backup

We've found a way to use batch files to back up data automatically at any time. You can also create a reminder, which indicates on which day you want to back up data, that will automatically appear on the screen on the date specified.

In order to do this, several files are needed. The following batch files are written using a C:\BAT directory. You should copy these files into a directory, called C:\BAT, and expand the path specification to include this directory. All the batch files are contained on this book's companion diskette, see Appendix E for more information about the companion diskette. The batch files work under MS-DOS 5.0 and consist of the following files:

✎ If you're using MS-DOS commands exclusively and something goes wrong, carefully look at the screen output of the individual commands.

1. ENTER: This file contains only a CR/LF (Carriage Return/Line Feed), which simulates pressing Enter for the DATE command, which waits for this key to be pressed before preceding. Create this file in MS-DOS in the following way:

The ENTER file

```
Type:

COPY CON \BAT\ENTER
Enter                                   (Press the Enter key once)
Ctrl + Z                                (or F6)
Enter
```

2. The following batch file, AUTO.BAT, redirects the output of the resident MS-DOS DATE command (the day's date) to the DAY.BAT file. Then it calls this batch file DAY.BAT.

The AUTO.BAT batch

```
@REM AUTO.BAT - by Manfred Tornsdorf
@REM Look for a note on a day
@REM Output a message for every day
@DATE < \BAT\ENTER > \BAT\DAY.BAT
@DAY
```

3. A batch file named CURRENT.BAT, prevents MS-DOS from displaying the error message, "Bad command or filename". After executing AUTO.BAT, the DAY.BAT file could have the following lines:

DAY.BAT: for MS-DOS 5.0

```
Current date is Tue 08-20-1990
Enter new date (mm-dd-yy):
```

MS-DOS tries to interpret the first word of the first line (Current) as a command and execute it. To prevent an error message, we need to create a batch file, called CURRENT.BAT:

The CURRENT.BAT batch

```
@CALL REMARK.BAT %4
@ECHO For accomplishing the daily tasks for [%3] press any key
@ECHO Abort with CTRL + C
@PAUSE >NUL
@%3.bat
```

4. The files DAY.BAT and CURRENT.BAT are necessary because CURRENT.BAT stores the day and date as parameters.

```
Current date is Tue 08-22-1990
```

The third parameter is the day of the week and the fourth parameter is the date.

CURRENT.BAT now calls the REMARK.BAT batch file with the third parameter ("%3"). This parameter contains the current date. REMARK.BAT can search for this date in a file named REMARK.DOC:

The REMARK.BAT batch

```
@ECHO I am looking for notes for the [%1]
@FIND "%1" \REMARK.DOC
```

5. In order for the search to be successful, there must be a file named REMARK.DOC in the root directory, not the BAT directory. This file should contain a paragraph for each item, which contains the correct date as displayed on the screen by the MS-DOS DATE command. The following is an example of what a REMARK.DOC would look like:

The REMARK.DOC file

```
05-22-1991: At 11:00 take your car for checkup
05-22-1991: At 2:00 PM appointment with Mr. Miller

06-04-1991: Time for a backup using D-BACKUP.BAT
REM This file should be in the root directory
```

6. After calling REMARK.BAT, CURRENT.BAT calls a batch file named for the day of the week. You should create batch files named for every weekday. The second parameter ("%3") passed to CURRENT.BAT contains three characters for the weekday ("Mon", "Tue", etc.); "%3.bat" calls one of these batch files (e.g., MON.BAT or TUE.BAT). These batch files can display messages by using ECHO or start programs.

7. Set up batch files for Sunday through Saturday (i.e., SUN.BAT through SAT.BAT) in the batch directory (C:\BAT). If nothing is planned for a particular day, simply write a line beginning with ECHO:

```
@ECHO No tasks for Tuesday.
```

For example, if you want to back up all changed document files and subdirectories from the C:BAT directory onto diskettes, use the D-BACKUP.BAT file (Section 6.4) in the MON.BAT file, as follows:

The activity file

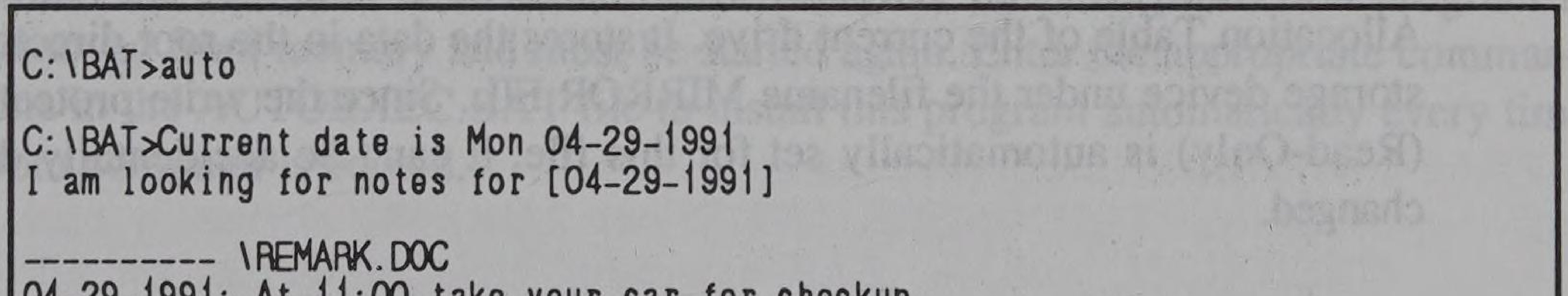
```
@REM Assignments for Mondays
@ECHO Today is Monday - Time for data security
@ECHO Insert data backup diskette into drive A
@ECHO Press space bar to continue, press CTRL + C to abort
@PAUSE > NUL
@CALL BACKUP C:\DATA\*.DOC A: 4-29-1991/S
```

8. To run these files, which contain notes for any date and any activities for every day of the week, each time your computer is switched on, add the following line at the end of your AUTOEXEC.BAT file:

```
@AUTO
```

This command line should be the last line in the AUTOEXEC.BAT file. If any command lines are placed after the call for AUTO, they won't be executed.

Here is an example display after running the AUTO.BAT file:



```
C:\BAT>auto

C:\BAT>Current date is Mon 04-29-1991
I am looking for notes for [04-29-1991]

---------- \REMARK.DOC
04-29-1991: At 11:00 take your car for checkup

C:\BAT>
For accomplishing the daily tasks for [Mon] press any key
Abort with CTRL + C
Today is Monday - Time for data security
Insert data backup diskette into drive A:
Press space bar to continue, press CTRL + C to abort
Reading source file(s)...
C:\BAT\REMARK.DOC
        1 File(s) copied
Finished!

C:\BAT>
```

6.7 FORMAT And DEL Or ERASE

Until MS-DOS 4.01, the FORMAT command completely removed the data from floppy diskettes and created a new system area (root directory, File Allocation Table (FAT)) on hard drives. As a result, the data on a diskette was permanently deleted and the data on the hard drive could no longer be accessed.

But now, when diskettes are formatted with MS-DOS 5.0, the existing data is no longer overwritten. The FORMAT command only checks the individual sectors of diskettes and hard drives for readability and then creates a new root directory and a new File Allocation Table for an empty storage device.

Protecting the root directory and File Allocation Table

You can protect your hard drive from accidental formatting by using the new MIRROR command. This command sets up a mirror image of the system areas that are overwritten by FORMAT. The UNFORMAT command uses this mirror image to restore the files on the storage device.

To back up the system information of a drive, call MIRROR with the appropriate drive letter:

```
MIRROR Drive
```

For example, to back up the system information of drive C:, use the following:

```
MIRROR C:
```

The MIRROR.FIL file

If a disk drive label isn't used, MIRROR backs up the root directory and File Allocation Table of the current drive. It stores the data in the root directory of the storage device under the filename MIRROR.FIL. Since the write protect attribute (Read-Only) is automatically set for this file, it can't be accidentally deleted or changed.

Protecting the partition table

Simply protecting the root directory and File Allocation Table isn't sufficient. If the partition table is destroyed, access to the hard drive could be blocked. A special switch for MIRROR saves the hard drive partition information onto a diskette.

```
MIRROR /partn
```

When this line is entered, a prompt, asking for the letter of the drive to which the partition information should be saved, appears. The default setting is drive A:. The command creates a special PARTNSAV.FIL file that the UNFORMAT command can use to restore the partition table to the hard drive.

Preventing the accidental deletion of files

Besides the protection methods described above, MIRROR also increases the chances that files which were accidentally deleted will be restored. A special option allows a TSR (Terminate and Stay Resident) program to be loaded into memory. The TSR program stores the location and name of each deleted file in a file named PCTRACKR.DEL, which is located in the root directory of the disk from which the file was deleted.

To protect your drive with this method, specify the /tDrive letter switch (without a colon) after MIRROR. In order to track deletions on more than one disk drive, use the following command line:

```
MIRROR /TC /TD /TE
```

File deletions on disk drive C:, D: and E: will now be tracked and an image of the system area will be created.

To create a mirror image of the system information for the hard drive (C:) and record all file deletions for the current work session on drive C: and A:, use the following command line:

```
MIRROR C: /tc /ta
```

To restore an accidentally deleted file, UNDELETE determines whether PCTRACKR.DEL exists and uses the information stored in it.

When the computer is switched off or restarted, the TSR protection program is removed from memory and must be started again. Enter the appropriate command line in the AUTOEXEC.BAT file to install this program automatically every time your computer is booted.

6.8 Recovering From An Accidental Format

MS-DOS 5.0 formats diskettes and hard drives so that, in case of an emergency, they can be restored with a new command, called UNFORMAT.

More information about preparing diskettes and hard drives can be found in Chapter 3 "Storage Devices".

To undo the formatting of a storage device, call UNFORMAT with the following storage device label:

```
UNFORMAT Disk_drive
```

Any changes made since the formatting occurred are lost. Since you may not be able to restore the storage device without any defects, only use this option in emergencies.

If the storage device was formatted with the special /U (unconditional format) switch of the FORMAT command or if there wasn't enough space to back up the system information, the restoration may be impossible.

UNFORMAT with backed up system area

Using MIRROR or FORMAT to back up the system areas of the storage device before formatting increases your chances of undoing the formatting.

If you accidentally formatted your hard drive (C:), return it to its original state with the command line:

```
UNFORMAT C:
```

UNFORMAT checks the storage device for system information backed up with either MIRROR or FORMAT. Since MIRROR creates a backup file of MIRROR.FIL with the .BAK extension, UNFORMAT might ask whether it should use the last (L) version or the prior (P) version. Usually the last version (L) should be used.

If UNFORMAT is used for a disk drive, you're prompted to insert the formatted diskette first.

Since using UNFORMAT incorrectly can destroy data, the process can be aborted by pressing Esc. However, before using UNFORMAT remember that this command can cause as much damage as accidentally formatting a storage device.

Using UNFORMAT without backed up data

If backed up system information doesn't exist or if the information is too old to be used for data restoration, UNFORMAT can be used without MIRROR.FIL. Specify the /U switch. However, you probably won't be able to perform a complete restoration:

```
UNFORMAT Disk_drive /U
```

After requesting a confirmation, UNFORMAT searches the entire storage device for existing information (files and directories). After completing the search phase, another prompt asks whether you're sure you really want to unformat the disk. Only then does UNFORMAT make any changes to the storage device.

One interesting option for UNFORMAT is that a test run, which only simulates making the changes to the storage device, can be performed. Since this procedure doesn't actually make any changes, your data isn't in any danger. Use the /TEST switch instead of the /U switch:

```
UNFORMAT Disk_drive /TEST
```

UNFORMAT switches

/J	Verifies that a file created by MIRROR has been saved and that it agrees with the system information on the disk. No other switch may be used with this switch.
/U	Unformats a disk without using the MIRROR file.
/L	Lists every file and subdirectory when used without the /PARTN switch. When used with /PARTN, UNFORMAT displays the partition table of the current drive.
/TEST	Simulates how UNFORMAT would restore the information on the disk without actually performing UNFORMAT.
/P	Sends output to the printer on LPT1:.
/PARTN	Restores a corrupted partition table of a hard disk. This switch requires PARTNSAV.FIL, which was created by using MIRROR with the /PARTN switch.

6.9 Recovering Files From Defective Storage Media

Occasionally MS-DOS can recover data from defective storage media.

Tools for creating backups and for recovering data (e.g., PC Tools, Norton Utilities or BeckerTools for Windows), which are more efficient than the tools provided by MS-DOS, are available. So before using the MS-DOS tools, investigate other options.

Buying an extra tool for creating backups is sensible because most backup programs require fewer diskettes and less time than the MS-DOS tools.

The MS-DOS RECOVER command can be used to recover files containing defective parts (sectors). As much of the file as possible will be recovered.

Before using RECOVER, back up any existing data. RECOVER changes data, which could cause data loss.

Restoring a file

To recover as much of a file as possible, use the command line:

```
RECOVER File
```

RECOVER indicates how much of the file can be recovered.

You should immediately back up the file and check it for readability. For example, if the file contains text, load it in the word processor and determine whether the complete file was recovered or if the file contains strange characters.

The RECOVER command can restore several files simultaneously by using the appropriate wildcards. Use the following command line to recover all of the files in the root directory of drive A:

```
RECOVER A:\*.*
```

Recovering all of the files of a drive

If you can no longer read the root directory of the drive correctly, you won't be able to restore the files as described above because RECOVER wouldn't find any files. If this happens, use RECOVER on the drive and try to restore as much as possible:

```
RECOVER Disk_drive
```

After you confirm a security prompt, the command restores as many files as possible. The command renames all of the files as FILEXXXX.REC, assigning them consecutive numbers (XXXX) and placing them in the root directory.

RECOVER changes the filenames of the specified files and divides defective files into several partial files. To a certain extent, this makes your data useless. So RECOVER should only be used when absolutely necessary. Also, you should always back up existing files.

If files were in subdirectories before you used RECOVER, they are restored to the root directory. The subdirectories are lost and appear as files in the root directory.

Example

A diskette contains the files Text1 to Text4 and the files Data1 to Data4 in a subdirectory called \DATA. CHKDSK provides the following information about the diskette:

```
Volume TEST-360K created 08-22-1990  5:36p
Volume serial number: 1629-16E9

    362496 bytes total disk space
      1024 bytes in 1 directory
    189440 bytes in 8 user file(s)
    172032 bytes available on disk

      1024 bytes in each allocation unit
       354 allocation units on disk
       168 available allocation units on disk

    655360 total bytes memory
    538944 bytes free
```

Suddenly the data on the diskette is no longer available because the directory was destroyed. The DIR command displays the following information:

```
Volume TEST-360K created 08-22-1990  5:36p
Volume number: 1629-16E9

        0 File(s)    172032 bytes free
```

Apparently files no longer exist. However, the low amount of free storage area indicates that the data wasn't deleted. Using RECOVER produces the following results:

```
Volume TEST-360-KByte created 08-22-1990  5:36p
Volume number: 1629-16E9

FILE0001 REC     11776  08.22.90  15.47
FILE0002 REC     10240  08.22.90  15.47
FILE0003 REC     92160  08.22.90  15.47
FILE0004 REC     47104  08.22.90  15.47
FILE0005 REC      1024  08.22.90  15.47
FILE0006 REC      6704  08.22.90  15.47
FILE0007 REC      4672  08.22.90  15.47
FILE0008 REC      8736  08.22.90  15.47
FILE0009 REC      5680  08.22.90  15.47
         9 File(s)    172032 bytes free
```

The files FILE0001.REC to FILE0004.REC are the first four files from the former root directory. FILE0005.REC is the subdirectory DATA, which has been converted to a file, and the files FILE0006.REC to FILE0009.REC were originally in the former DATA subdirectory.

As you can see, the sequence of the files can be used to trace their origin. If you don't know the sequence, the only criterion you can use is the size of the file.

Since RECOVER restores all of the files to the root directory, which on a 360K diskette can only contain 111 files, it's possible that this limit could be exceeded. RECOVER will abort with a message informing you of this.

On a different storage device, back up the files you've recovered up to this point and delete them from the defective diskette. Then RECOVER can restore more files.

6.10 Restoring Deleted Files

Not only can files be created and edited on a computer, but one or more files can also be deleted in fractions of a second. Because of this, eventually you may accidentally delete a file.

However, usually when a file is deleted it isn't really "erased". So, many times a deleted file can be recovered. One way you can tell that the computer doesn't really "erase" a file (i.e., permanently delete it) is by the speed of the deletion process, which is fast even for large files. If the computer permanently deleted the file, the process would take as long as it does to copy a file.

How does MS-DOS actually delete a file?

A file consists of two very different elements: information stored in the directory of the diskette or hard drive and the actual data contained in the file. This data is stored in special clusters.

A cluster is the smallest unit in which data can be stored on a storage device. In a special area, called the File Allocation Table (FAT), MS-DOS also stores information about which areas of the storage medium are used.

When a file is deleted, two things occur:

Only the entry is deleted

The first character of the filename in the directory is deleted and replaced by the ASCII character 229 (hexadecimal, E5), which displays the "σ" character. However, since MS-DOS can't display file entries that have been changed this way, a special utility program is needed. So if you deleted a file named COMP0001.DOC and used the appropriate utility programs to look at the file entry, you would see "σOMP0001.DOC".

Memory area is designated as free

The units of memory occupied by the file (clusters) are marked as free in the FAT. This allows MS-DOS to use these clusters for storing new data. However, the data on the diskette isn't overwritten until you actually store new data.

There are two ways to destroy a file permanently:

- As soon as MS-DOS places a file in the directory, it sets up a new file entry. This overwrites and destroys the file entry of the deleted file. At this point, all of the information about this file is lost (name, extension, date, size and starting address of the data on the storage device).

- When data is stored on the storage device and more space is needed, the storage units previously assigned to the deleted file are used. As soon as MS-DOS stores new data in these areas, the original data is lost.

Because of both of these possibilities, don't make any changes to the contents of your disk after accidentally deleting one or more files. This could prevent the file from being restored.

This destruction of data can be recovered in two steps. First, use a special program to remove the "σ" symbol from the filename and replace it with any normal character. The recovery program may prompt the user for the missing character in the filename.

The second step is to reallocate the cluster in the FAT. The cluster released by MS-DOS must be joined to a chain and designated as used.

Usually it's possible to undo the deletion of a file automatically if nothing else has been stored on the storage device since the deletion. The deletion can't be undone if the first specified cluster of the file in the directory is used by another file or if the file is stored in several "cluster chains" instead of contiguously on the storage device.

If the program is stored in several chains, the utility programs used to recover files can no longer determine which clusters belonged to the deleted file. Unfortunately you must determine which clusters belong to the deleted file in order to undo the deletion of a file.

Undeleting files

If no changes have been made to the storage device after one or more files have been accidentally deleted, you can usually undo the deletion.

Use UNDELETE to recover the deleted file if you haven't saved anymore files since deleting the file(s). This command makes the file available again. To do this, call UNDELETE followed by the filename of the deleted file.

If a filename isn't entered after UNDELETE, the command acts on all of the files in the current directory on the current drive.

If several files were deleted in a directory, undo the deletion by calling UNDELETE in that directory and confirming the recovery of each file. Enter the first letter of the filename to recover it. When this is completed, the command indicates whether or not the file was recovered.

The first letter of the filename must be entered because it was destroyed by the deletion process. You don't have to use the same first letter that was previously used.

Creating a deletion list

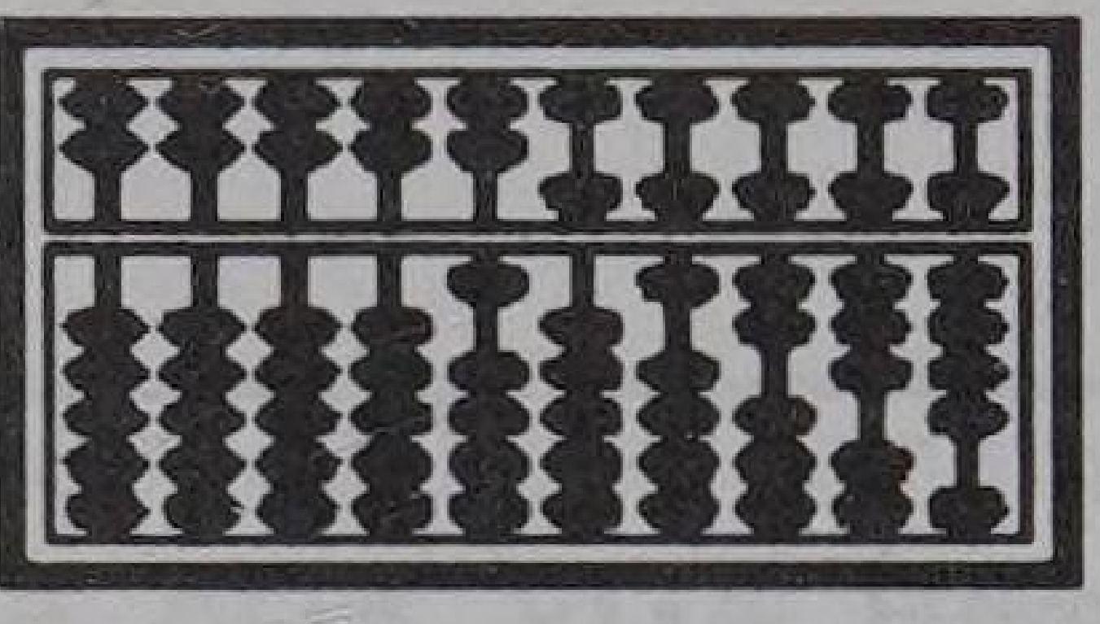
To further protect your files from accidental deletion, document each deletion in a list. To do this, call MIRROR with the /T switch and specify the letter of the drive for which you want to create a deletion list.

With the help of this information, UNDELETE can reconstruct deleted files more efficiently. To make a deletion list for drive C:, type the following command line each time you re-start the computer:

```
MIRROR C: /TC
```

This line could also be included in the AUTOEXEC.BAT file.

Chapter 7



Application Programs

Whether you realize it or not, application programs rely on MS-DOS to perform many of their tasks. For example, to format a diskette, a word processor will call on the services of MS-DOS rather than perform this function itself.

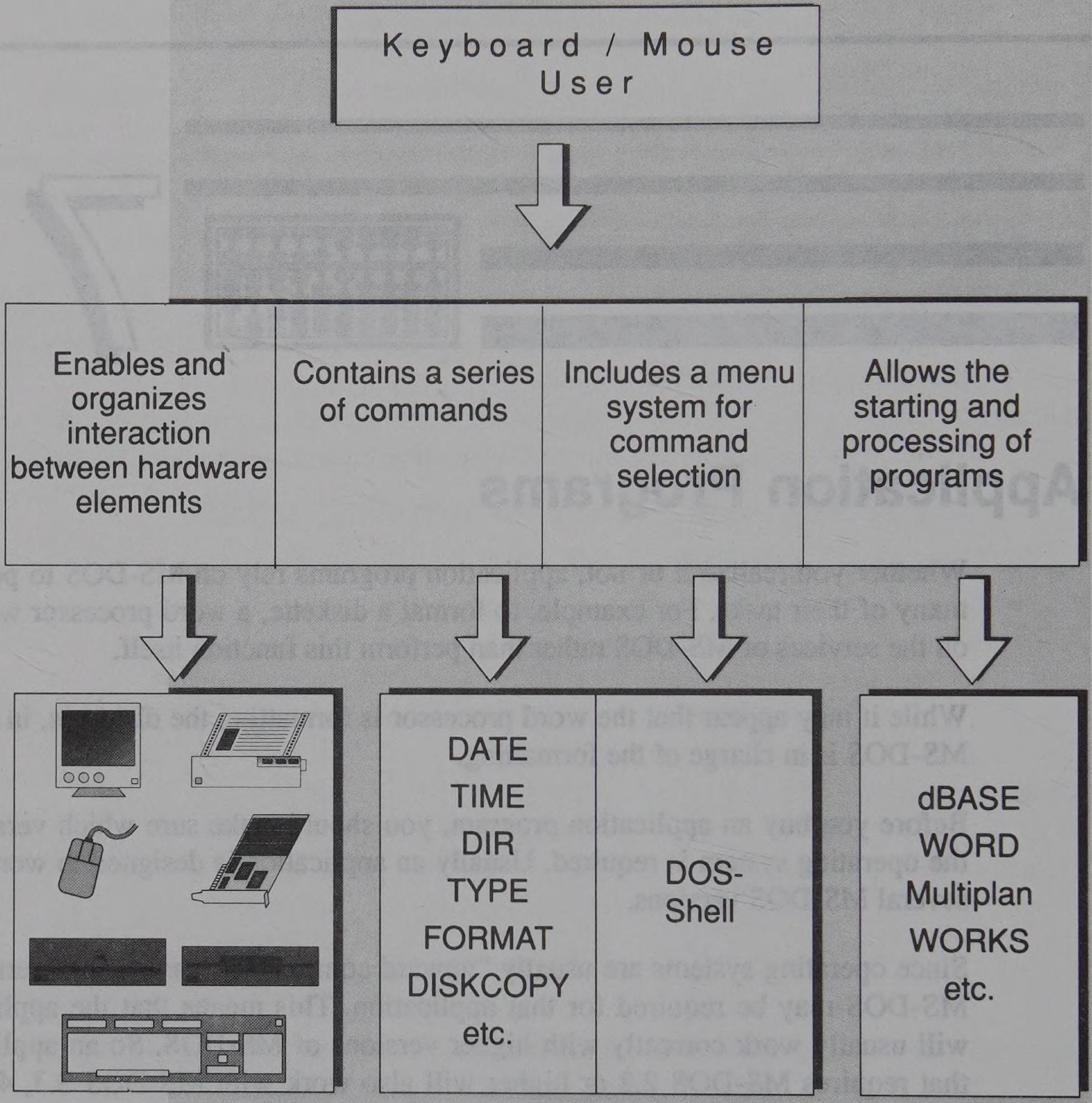
While it may appear that the word processor is formatting the diskettes, in reality MS-DOS is in charge of the formatting.

Before you buy an application program, you should make sure which version of the operating system is required. Usually an application is designed to work with several MS-DOS versions.

Since operating systems are usually "upward compatible" a minimum version of MS-DOS may be required for that application. This means that the application will usually work correctly with higher versions of MS-DOS. So an application that requires MS-DOS 2.2 or higher will also work with MS-DOS 3.3, 4.0 and 5.0.

However, there are exceptions to this rule. Some special utilities or tools often depend on specific internal functions of the operating system. For example, a utility program might work with MS-DOS 3.3, but not with MS-DOS 5.0. If you're not sure which version is required, ask your computer dealer or the manufacturer of the program.

The following figure shows the relationship of the operating system and an application program. It demonstrates the most important and basic link between the user's idea or task and the finished product. Cooperation between the operating system and the application program is extremely important.


Keyboard / Mouse
User
Enables and organizes interaction between hardware elements
Contains a series of commands
Includes a menu system for command selection
Allows the starting and processing of programs
DATE
TIME
DIR
TYPE
FORMAT
DISKCOPY
etc.
DOS-Shell
dBASE
WORD
Multiplan
WORKS
etc.

The operating system is important to how your computer is used

When a user begins to work on a task, he/she has an idea of how to complete it. The user is already using many of the diverse capabilities of the operating system when the computer is switched on and the application program is started.

Using the services of the operating system, an application program (such as a word processor) is used to produce the desired end result (for example, a letter).

When working with files, application programs usually use many of the internal functions of the operating system (DOS). For example, to copy a document file to a diskette or back up your hard drive, the application program calls on the various functions of DOS to perform these.

7.1 Installing Programs

Most application programs are distributed on floppy diskettes. These programs usually contain many files on several diskettes. To use an application program, these files must be copied into a specific directory on your hard drive. This is the only way to control and maintain programs. We advise that you avoid keeping several different programs in a single directory.

Copying files isn't always enough

Computer systems vary greatly in hardware and performance. Some systems use different types of graphics adapters (CGA, EGA, VGA, Hercules, etc.) and may or may not include a hard drive or mouse or other hardware.

Many application programs must be adapted or customized to the specific hardware components of your particular computer system. So simply copying all of the files from the diskettes onto your hard drive may not be enough. Many application programs must be *installed* before they will operate properly.

Interactive installation programs

Most application programs include special installation programs. These programs copy and configure the files on your hard drive. Some installation programs only tell you what they are doing and prompt you to change diskettes. Other installation programs work with you. This means that the installation program prompts you for various information so that it can determine the specifications required for optimum performance with your computer system.

Why use installation programs?

In many cases it's important that the application program determine which graphics adapter (graphics card) your computer uses. This is especially important for graphics programs. You may also have to answer questions about the operation of your mouse (mouse type, serial, bus, etc.), the kind and model of your printer or other accessories and the configuration of your computer. The application program records the information and adapts itself to function properly with your system. This information is usually stored in a special file created during the installation process. If this file is missing, your application program either won't run or the program will crash, which causes loss of data.

Installation programs are also necessary for another reason. Some applications are saved on the diskette in a special compressed form to save space. The information is compressed by using special techniques. These files are sometimes called "packed" files.

The advantage of packed files is that more data can be stored on a diskette. However, these files require a program to "unpack" the files to their original size.

Installation programs perform this function automatically during the installation process. For example, the files on your MS-DOS 5.0 diskettes were compressed before being placed on the diskettes and uncompressed when installed to your hard drive. It would be difficult and time-consuming for users to unpack these files by hand.

Before installing a program, always make backup copies of the original diskettes and use the backup copies for installation. Also, be sure to write protect the original diskettes before making the copies. This will insure that the original diskettes aren't destroyed by a copying error.

Installing an application

If the application program you want to install comes on several diskettes, one of the diskettes is usually labelled "Start disk", "Installation" or something similar. The installation program, if included with the application, is usually on this diskette. The most common names for installation programs are INSTALL or SETUP. You can use `DIR` to display the existing files and programs or type:

```
DIR *.EXE

DIR *.COM

DIR *.BAT
```

to display all of the executable files and determine the name of the installation program. You can also refer to the program's user manual for instructions on installing the application. The installation instructions are usually located in a chapter near the front of the manual or in a separate booklet packaged with the program.

Using a a batch file can simplify searching for executable programs on your disks. The following batch program, SHOWPRG.BAT, displays all of the executable files, in the current directory, with the extensions .COM, .EXE or .BAT. The batch file displays the programs in alphabetical order and page by page:

```
@ECHO OFF
REM SHOWPRG - Display all programs
IF EXIST *.EXE DIR *.EXE  >FP.TMP
IF EXIST *.COM DIR *.COM  >>FP.TMP
IF EXIST *.BAT DIR *.BAT  >>FP.TMP
FIND <FP.TMP /V "e" | SORT | MORE
IF EXIST FP.TMP DEL FP.TMP
ECHO ON
```

Save this batch file in a directory that has been specified in your search path. This lets you start the batch file from any directory. To display all of the programs on a diskette in drive A:, set A: as the current drive and call the batch program:

```
A:
SHOWPRG
```

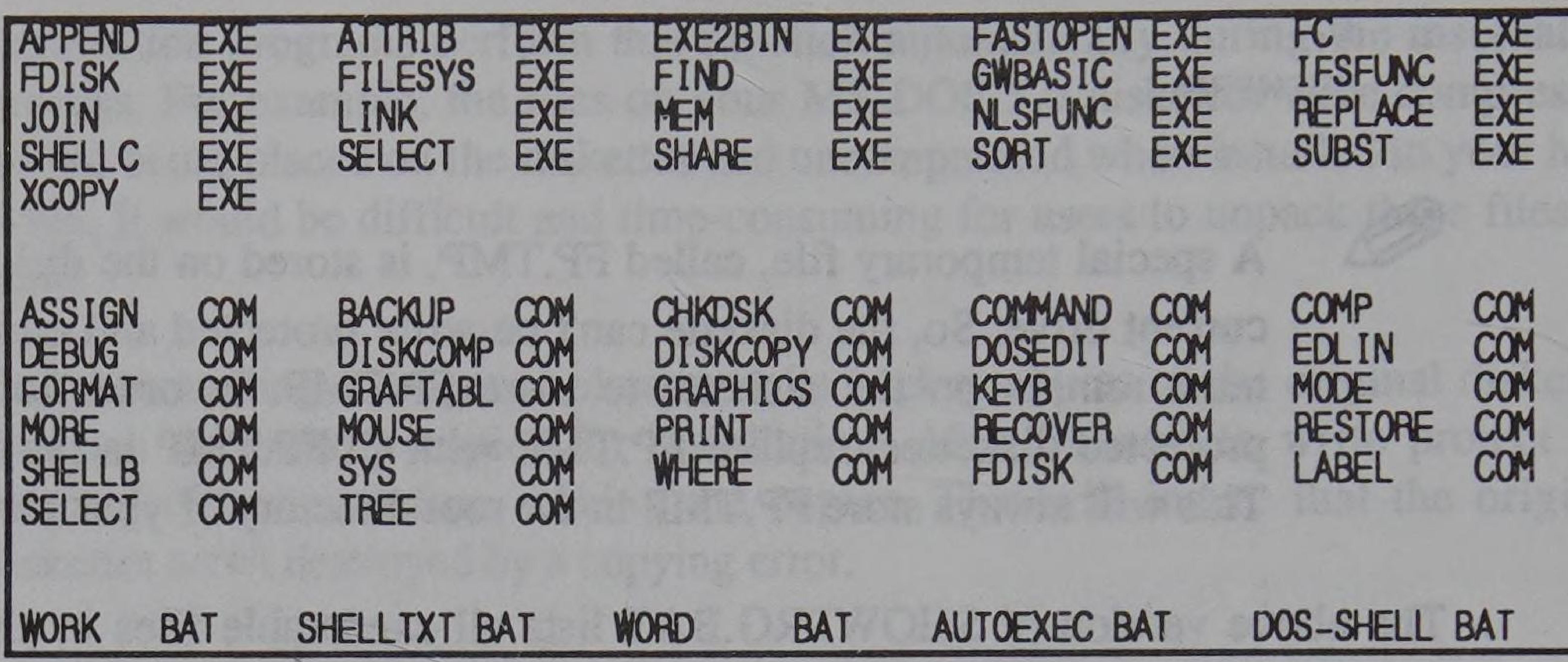
A special temporary file, called FP.TMP, is stored on the diskette of the current drive. So, the diskette can't be write protected and enough space must remain on the diskette to store FP.TMP. In order to use write protected diskettes, replace FP.TMP with C:\FP.TMP in the batch file. This will always store FP.TMP in the root directory of your hard drive.

The above version of SHOWPRG.BAT lists all executable files in alphabetical order. Frequently, users prefer to have the files displayed in a different order. For example, the files can be listed by extension (.COM, .EXE or .BAT) with as many files on the screen as possible. To do this, modify SHOWPRG.BAT as shown below:

```
@ECHO OFF
IF EXIST *.COM DIR *.COM /W >FP.TMP
IF EXIST *.EXE DIR *.EXE /W >>FP.TMP
IF EXIST *.BAT DIR *.BAT /W >>FP.TMP
FIND <FP.TMP /V "e" | MORE
IF EXIST FP.TMP DEL FP.TMP
ECHO ON
```

If you don't want to write to a floppy diskette, simply replace FP.TMP with C:\FP.TMP in the batch file.

This version of the batch program displays all the executable files in five columns on the screen. First the COM files are displayed, followed by the EXE files and finally the BAT files. MS-DOS follows this sequence when it searches for executable files. This version also excludes all of the information about the storage device, which provides more space for the files. The following is an example of what the display could look like:



```
APPEND   EXE     ATTRIB   EXE     EXE2BIN  EXE     FASTOPEN EXE     FC       EXE
FDISK    EXE     FILESYS  EXE     FIND     EXE     GWBASIC  EXE     IFSFUNC  EXE
JOIN     EXE     LINK     EXE     MEM      EXE     NLSFUNC  EXE     REPLACE  EXE
SHELLC   EXE     SELECT   EXE     SHARE    EXE     SORT     EXE     SUBST    EXE
XCOPY    EXE

ASSIGN   COM     BACKUP   COM     CHKDSK   COM     COMMAND  COM     COMP     COM
DEBUG    COM     DISKCOMP COM     DISKCOPY COM     DOSEDIT  COM     EDLIN    COM
FORMAT   COM     GRAFTABL COM     GRAPHICS COM     KEYB     COM     MODE     COM
MORE     COM     MOUSE    COM     PRINT    COM     RECOVER  COM     RESTORE  COM
SHELLB   COM     SYS      COM     WHERE    COM     FDISK    COM     LABEL    COM
SELECT   COM     TREE     COM


WORK     BAT   SHELLTXT BAT     WORD     BAT     AUTOEXEC BAT     DOS-SHELL BAT
```

The executable files

Sometimes a file called README (or something similar) is located on one of the diskettes (usually on diskette no. 1). These files contain information that was not included in the manual. Perhaps it has information that was inadvertently omitted when the manual was printed. Sometimes you'll find information about the installation program in a README file. So you should always read this file before beginning the installation.

README files are text files that can be read with an editor or a word processor. If you don't have either of these programs on your computer, you can read this information file by using the TYPE command. However, entering TYPE README.DOC will cause the entire contents of the file to race over the screen. It's easier to read the file if you add the MORE command. (You can create the pipe symbol by pressing Alt + 1 + 2 + 4, if your keyboard does not have this character.) The command is as follows:

```
TYPE README.DOC | MORE
```

If the file doesn't have an extension (.DOC, .TXT), you don't need to enter one. Whenever the text fills the screen, the output stops until a key is pressed. After pressing a key, the next page of text is displayed.

With MS-DOS 5.0, reading files is even easier. You can use the MS-DOS Editor program, EDIT, to display files. Type the following command:

```
EDIT README.DOC
```

The file will be loaded and you can use the cursor keys and the PgUp and PgDn keys to scroll through the document. You can even print the document by selecting the **Print** command from the **File** menu.

Installing application programs properly

There are several factors to consider when installing application programs. By experimenting, you could probably install programs onto your hard drive without using the installation programs. However, this method could eventually lead to problems. One of your programs might suddenly stop working and you won't know what caused the problem. The following are a few simple rules for installing application programs.

Directories for program files

Always use the installation program supplied with your application software. This will ensure that the software is installed correctly and that maximum performance will be achieved. However, if you want to install application programs by hand (without the supplied installation program), always copy the files of these programs into their own subdirectories. For example, there would be a separate directory for Microsoft Word, a separate directory for dBASE, etc. Usually installation programs also create new directories and copy the files of an application program into them.

Separating the different programs and their files isn't the only reason new directories are created. Files of the same name can be used by different programs. So if the programs aren't in different directories, files can be overwritten each time you copy new program files onto your disk.

Also, if you decide to delete a program from your hard drive, you won't be able to determine which files belong to that particular application program. So placing all files in a common directory can eventually cause many problems.

Directories for data files

You should also create a separate directory for all of the files created by an application program. This will provide a better overview of the contents of your hard drive and will simplify backing up the data on your hard drive.

Since the files that the application program needs in order to operate usually aren't changed after installation, you only need to back up the contents of this directory once. The data that the application program processes, on the other hand, needs to be backed up at regular intervals. If the data is stored in a special directory, the backup process can be quicker, easier and more accurate.

If you use one application program to work on several types of data, you should store the data in their own subdirectories. This makes it easier to find files of the same type.

Although all files in a directory must have different names, files in different directories can have the same name. For example, if you do your taxes on your computer, you can set up a directory for each year and use the same filenames (STATE.TXT, FEDERAL.TXT, LOCAL.TXT).

Calling programs and data in one command

Many application programs allow you to specify which data file is to be loaded when the program is started. The application program starts and opens the data file. So you don't have to wait for the program to start before specifying which file to use.

To make the directory containing the data files the current directory for the application program, change to the data directory before starting the program and call the application program. Remember that the MS-DOS PATH command must be used to set the search path for the directory containing the program files.

The complete path to the program can also be specified. For example, to start Microsoft Works from the C:\WORKS directory, you would type:

```
C:\WORKS\WORKS
```

Obviously, you could also create a batch file, called WORKS.BAT, that would contain the above command line for starting Works.

The path to the program directory

If the application program is installed on your hard drive in its own subdirectory, the PATH command can be used to include this directory in the search path. Then you can start the program from any directory. If your search path for MS-DOS commands is C:\DOS, to include the Microsoft Word word processor in the search path, expand the PATH line in AUTOEXEC.BAT.

```
PATH C:\DOS;C:\WORD
```

If you used the Microsoft Word installation program, the search path is set and AUTOEXEC.BAT is modified during the installation process.

Avoid too many program directories in the search path. First, an MS-DOS command line can contain a maximum of only 127 characters. Since the complete search path must be specified, the entire path specification can be up to 122 characters long. Second, when a command is entered, MS-DOS searches the entire search path for the command. So the longer this search path is, the longer it takes MS-DOS to search it.

Only the programs that are required (dBASE IV, Word 5.0 etc.) or programs that are constantly used should be included in the search path. Instead, you can use a batch file to start programs that aren't used as often. By using a batch file, you don't have to change directories (see below). If you use the MS-DOS Shell, you could make the appropriate entries for these programs in the program list, which we'll explain later. This will let you start these programs easily.

The start batch solution

Creating a batch file that can call a program and load a file is easy. You don't even have to know a lot about batch programming. Simply create a one line ASCII file (a text file without formatting) that contains the complete path for starting the program and one or more variables for the names of files to be opened. These variables are used to pass the parameters from the batch file to the program.

Create a batch file called W.BAT for calling Word and place it in a directory which has been defined by PATH (for instance, the BAT directory discussed earlier). Use an editor program (Word or the MS-DOS Editor) to type the following line, then save it as an ASCII file:

```
C:\WORD\WORD %1 %2 %3
```

To start Microsoft Word to edit the LETTER.DOC file in the current directory (C:\DOC), type:

```
W LETTER
```

Avoiding problems during installation

As we previously mentioned, most application programs contain installation utilities. However, even with these programs, you can still encounter problems. By taking a few precautions, you'll be able to avoid most of these problems.

Automatically changing the start and configuration files

Some programs automatically modify AUTOEXEC.BAT and CONFIG.SYS during installation. Commands in these files, which the application program needs in order to function properly, are inserted in these files. Depending on the installation program, the original versions of AUTOEXEC.BAT and CONFIG.SYS are saved with new extensions (.SAV or .OLD, for example). Since you can't be certain whether you will still have copies of the original files after installation, use COPY to back up these files onto a diskette before running the installation program.

Use Chapter 18, The MS-DOS 5.0 Commands, and Appendix A, Device Drivers (the reference lists all of the MS-DOS commands that can appear in the AUTOEXEC.BAT or the CONFIG.SYS), to check the changes or refer to Section 2.1 "Setting Up MS-DOS", which provides a short description of all the important elements of these files.

Many installation programs make changes to three settings:

- They extend the search paths for commands (PATH) to the directory, in which the new application is installed.

- They check the BUFFERS parameter for a minimum value and, if necessary, increase this value. Although this speeds up access to the hard drive, it decreases the amount of memory proportionally. If you install a new application program and a previous one no longer functions correctly or an error message indicating that there isn't enough memory is displayed, the reason may be due to an increase in BUFFERS.

- They check the FILES parameter for a minimum value and, if necessary, increase the value. This sets the number of files that can be used simultaneously by MS-DOS. While in many cases the value 20 is sufficient, some programs (databases, for example) require a larger value. Although increasing the value for FILES results in less memory, don't undo the change the installation program has made.

Besides these changes, which generally don't affect other programs, there are changes that can cause problems.

Early versions of GEM (Graphics Environment Manager) automatically create a search path (SET COMSPEC) for the command interpreter. This can lead to problems if COMMAND.COM isn't in the specified directory. After quitting some programs you may receive the error message "Invalid command interpreter". If this happens, you'll have to reboot your computer. To solve this you can either undo the change or correct the GEM.BAT file, which changes the system variables.

Keeping older program versions

If you have an application program on your hard drive and purchase a new version of this program, don't replace the early version of the application program with the new one immediately. Install the new version so that both are available for a transitional period. Sometimes new versions of a program won't work with files that were created with earlier versions of the same program.

When installing a new version, use a new directory with a meaningful name. For example, if you have been using Microsoft Word 4.0, stored in the C:\WORD directory, and want to switch to Microsoft Word 5.0, install the new version of Word in a directory called C:\WORD5.

Some installation programs specify the name for the new directory. So you aren't able to change the name during installation. Because of this, you should change the name of the directory containing the earlier version before starting the installation program. This fools the installation program and prevents all of the old files from being overwritten.

Installing the new version in addition to the earlier one can help you avoid some potential problems:

- Sometimes the new version of the program can't use all of the data from the earlier version. If this happens, use the older version to process this data.

- The driver program supplied with the new program for the screen or printer could cause problems with your hardware. For example, you may have a full page monitor and a special driver program for the early version of the program. The new version might not be able to use this special driver.

- Some programs can only be installed once and have to be "deinstalled" onto diskettes with a special program before installing them on another computer. If a new version of the program is installed in the directory containing the early version, it overwrites important information and the old version can no longer be used (dBASE III plus).

Check your hard drive for the necessary storage space

When you install a new version of a program in addition to an earlier version, there may not be enough space on your hard drive. When this happens the installation program might terminate unsuccessfully. The program may also abort with a message, indicating that the hard drive is full. So remember that not all installation programs check for sufficient disk space on the storage device before starting.

The amount of storage space remaining on the device is displayed at the end of the DIR command display.

There should be enough space on the hard drive to copy all of the installation diskettes onto the hard drive. For example, if an application program is distributed on twelve 360K diskettes, you'll need at least 4.3 Meg of memory. If you don't have enough room on the hard drive, use XCOPY or BACKUP to move rarely used data to diskettes. First back up this data to a diskette, then delete the data from the hard drive.

Some programs require much more space on the hard drive than indicated by the number of distribution diskettes. One reason for this is that many programs come in packed formats. During installation, the files are automatically unpacked which therefore requires more space. Also, some programs require additional storage space on the hard drive. For example, Windows 3.0 uses the hard drive to store temporary data.

7.2 The Command Interpreter

Application programs can be started just like MS-DOS commands. For this, the computer must display the system prompt.

Loading application programs from a diskette

Insert the diskette containing the application program in drive A: and make this drive the current drive by typing:

```
A:
```

Enter the name of the application program and press Enter.

If the application program was installed on diskettes, usually a start or program diskette must be inserted to start the program.

Some applications are started from a batch file. For example, Ventura Publisher is started using VP.BAT. Use the following command to check whether this batch file is on the diskette:

```
DIR *.BAT
```

Some programs are loaded entirely in the computer's memory in such a way that you can replace the program diskette with a data diskette. If this isn't the case, insert the data diskette in the second drive (if you have one). Don't save data on the program diskette because there usually isn't enough unused disk space and you could also destroy the application program through an error.

Starting application programs from the hard drive

If PATH defines a path for the directory of the application or a batch file, the application can be started from any directory.

If a path wasn't defined, you must make the hard drive the current drive:

```
C:
```

Use the Change Directory command CD to specify the name of the directory in which the application program is located:

```
CD Directory
```

The application can then be started by typing the program name.

Many programs aren't started directly. Instead they use a supplied batch file, such as VP.BAT for Ventura Publisher. You can use the following command to check for this batch file in the directory of the program:

```
DIR *.BAT
```

Overlay files

Instead of being completely loaded in memory, some application programs use OVL files, which are loaded as needed. To start these application programs from any other directory, sometimes you may need to use the MS-DOS APPEND command, which sets a search path for programs and files. If this special search path isn't defined, the application program might not be able to load the OVL files and will end with an error message (and possible loss of data).

For example, to set a search path for program and OVL files for dBASE in the C:\DBASE directory, insert the following line in the AUTOEXEC.BAT:

```
APPEND C:\DBASE /X:ON
```

Searching for programs or files

If an application program is installed on the hard drive but you can't remember the name of the directory, use the new options of DIR to search the hard drive for the program file. If you can remember the name or part of the filename used to start the program, use DIR with the part you remember to search the entire hard drive. You can also substitute wildcards (* and ?) for the unknown parts of the name.

First use the CD\ command to change to the root directory of the storage device to search. The /S switch instructs DIR to include all subdirectories in the search. If you're currently in the root directory and include all subdirectories in the search, the entire storage device is searched. The command looks like this:

```
DIR Filename.extension /S
```

DIR displays information about the directory where the file was found as well as all the information usually displayed about a file.

Searching for programs: the easy way

You can create a batch file that will search the entire storage device for files. For simplicity's sake, name this file WHERE.BAT. By using this batch file, you don't have to change to the root directory and type DIR along with the /S switch. For example, if you type WHERE MOUSE.*, DIR will display all of the files called MOUSE, along with a line indicating in which directory the file is located. The following is the WHERE.BAT batch:

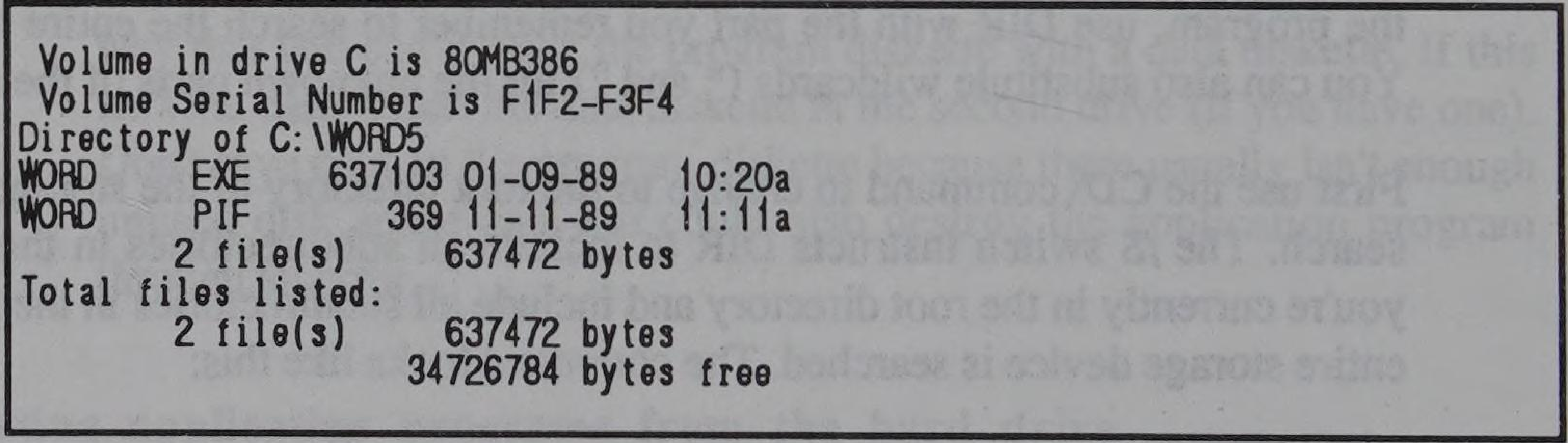
```
@REM WHERE.BAT Section 7.2
@REM Batch for searching the current drive for a file
@ECHO OFF
IF "%1" == "" GOTO SYNTAX
CD\
DIR %1 /S
GOTO End
:Syntax
ECHO Correct syntax is: WHERE FILENAME.EXT
:End
ECHO ON
```

If hard drive C: isn't the current drive, make it the current drive before using WHERE.

If you don't remember the name of the directory, in which the word processing program Word is stored and you're also not sure whether the program file is called WORD.EXE or WORD.COM, search for it by using WORD.*. Type the following:

```
WHERE WORD.*
```

The batch file searches the entire drive and displays the directory, in which each file was found. For this example with Word, the screen display should look like the following:



```
 Volume in drive C is 80MB386
 Volume Serial Number is F1F2-F3F4
Directory of C:\WORD5
WORD     EXE    637103 01-09-89   10:20a
WORD     PIF       369 11-11-89   11:11a
       2 file(s)     637472 bytes
Total files listed:
       2 file(s)     637472 bytes
                   34726784 bytes free
```

The program file is called WORD.EXE and is stored in the C:\WORD directory.

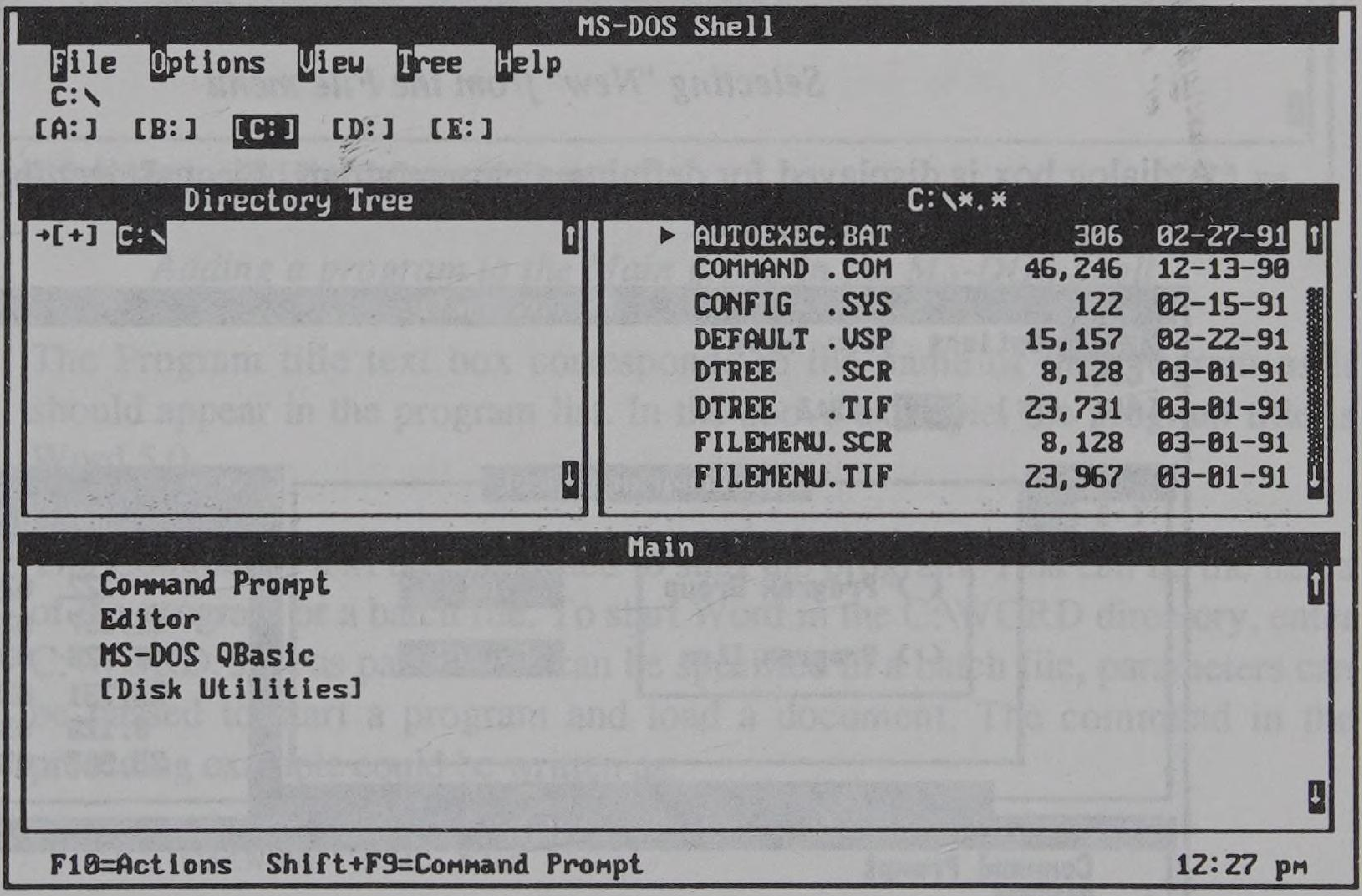
7.3 Starting Programs From The MS-DOS Shell

You can also start programs from the MS-DOS Shell. Using the MS-DOS Shell, there are three ways to start programs that are just as convenient as starting programs from the command line.

- Select a program from the main list.
- Select a program file or a file associated with the program from a file list.
- Select the **Run** command from the **File** menu.

The program list

If you use a particular program frequently, you should include it in the program list in either the Main group or create a new program list. These groups are displayed at the bottom of the default MS-DOS Shell screen.

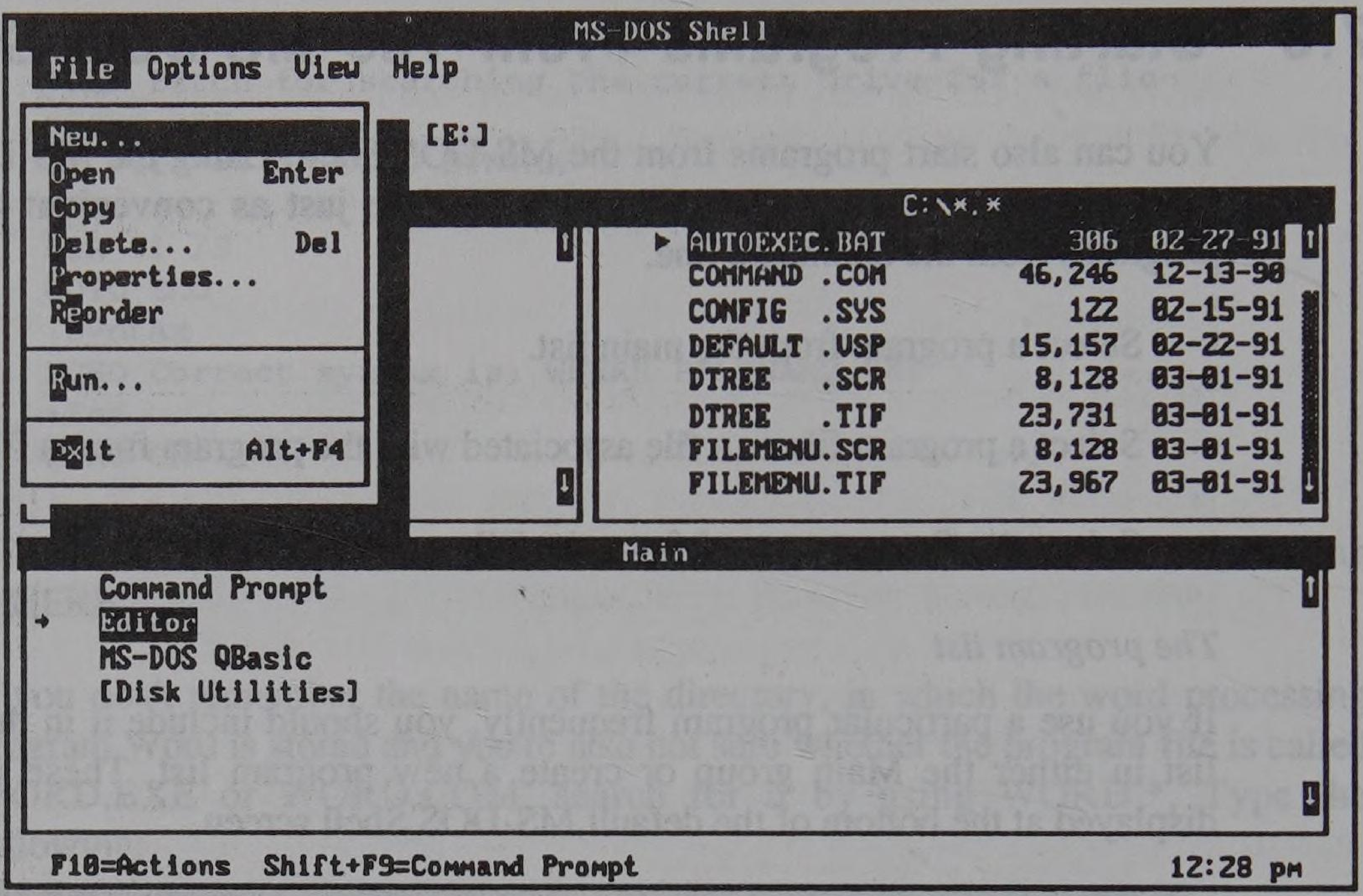

MS-DOS Shell
File Options View Tree Help
C:\
[A:] [B:] [C:] [D:] [E:]
Directory Tree
→[+] C:\
C:\*.*
Main
Command Prompt
Editor
MS-DOS QBasic
[Disk Utilities]
F10=Actions Shift+F9=Command Prompt
12:27 pm

```
▶ AUTOEXEC.BAT      306  02-27-91
  COMMAND .COM   46,246  12-13-90
  CONFIG  .SYS      122  02-15-91
  DEFAULT .VSP   15,157  02-22-91
  DTREE   .SCR    8,128  03-01-91
  DTREE   .TIF   23,731  03-01-91
  FILEMENU.SCR    8,128  03-01-91
  FILEMENU.TIF   23,967  03-01-91
```

The Main group of the MS-DOS Shell is in the lower part of the window

Select the Main group either by clicking the left mouse button or by pressing the Tab key until the first item (Command Prompt) is highlighted. Now you're ready to add a new item to the program list in the Main group.

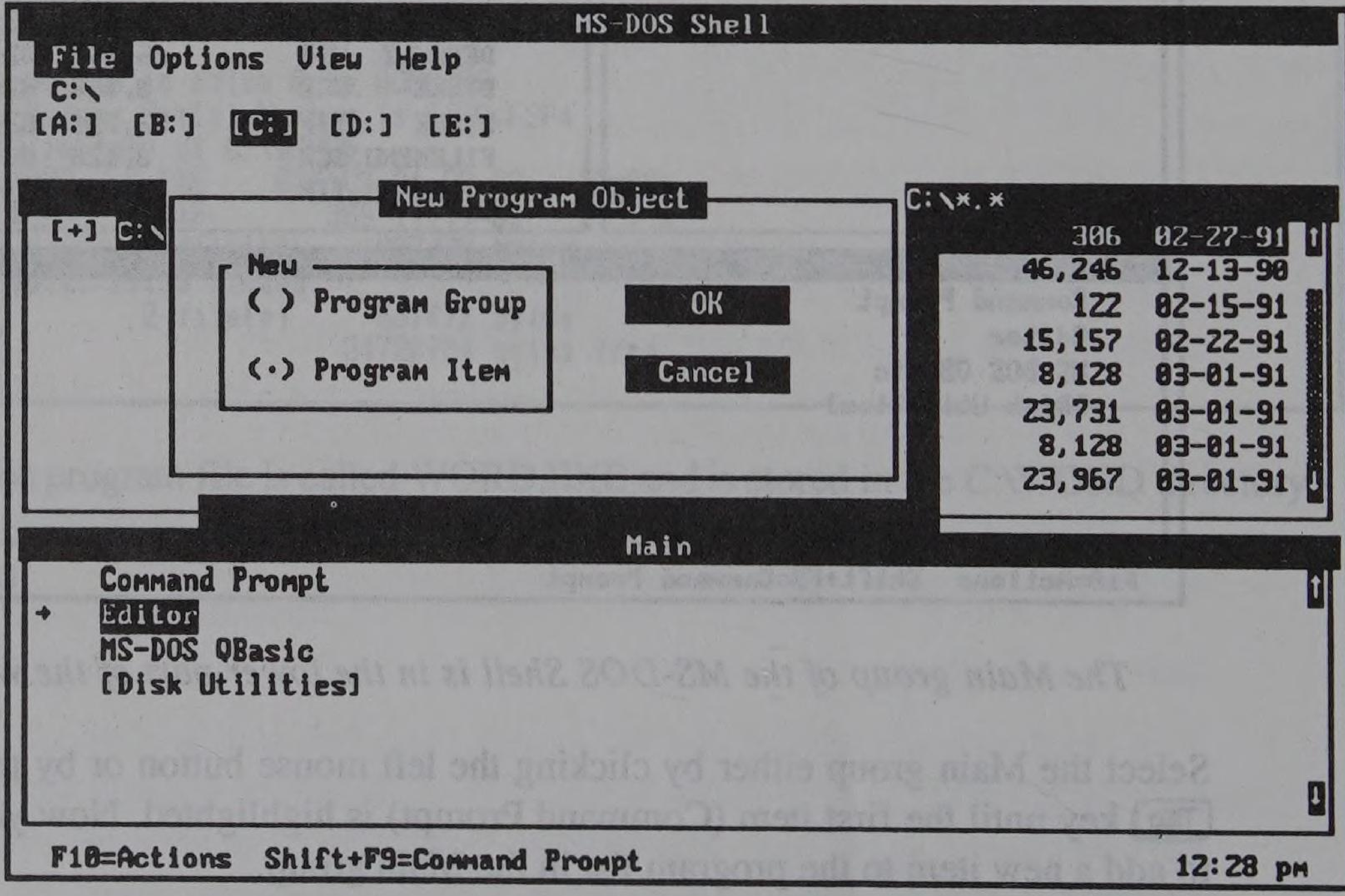
To add a program to the main list, activate the **File** menu and select the **New** command.


MS-DOS Shell
File
Options
View
Help
New...
Open Enter
Copy
Delete... Del
Properties...
Reorder
Run...
Exit Alt+F4
[E:]
C:\*.*
Main
Command Prompt
Editor
MS-DOS QBasic
[Disk Utilities]
F10=Actions Shift+F9=Command Prompt
12:28 pm

```
AUTOEXEC.BAT       306  02-27-91
COMMAND .COM    46,246  12-13-90
CONFIG  .SYS       122  02-15-91
DEFAULT .VSP    15,157  02-22-91
DTREE   .SCR     8,128  03-01-91
DTREE   .TIF    23,731  03-01-91
FILEMENU.SCR     8,128  03-01-91
FILEMENU.TIF    23,967  03-01-91
```

Selecting "New" from the File menu

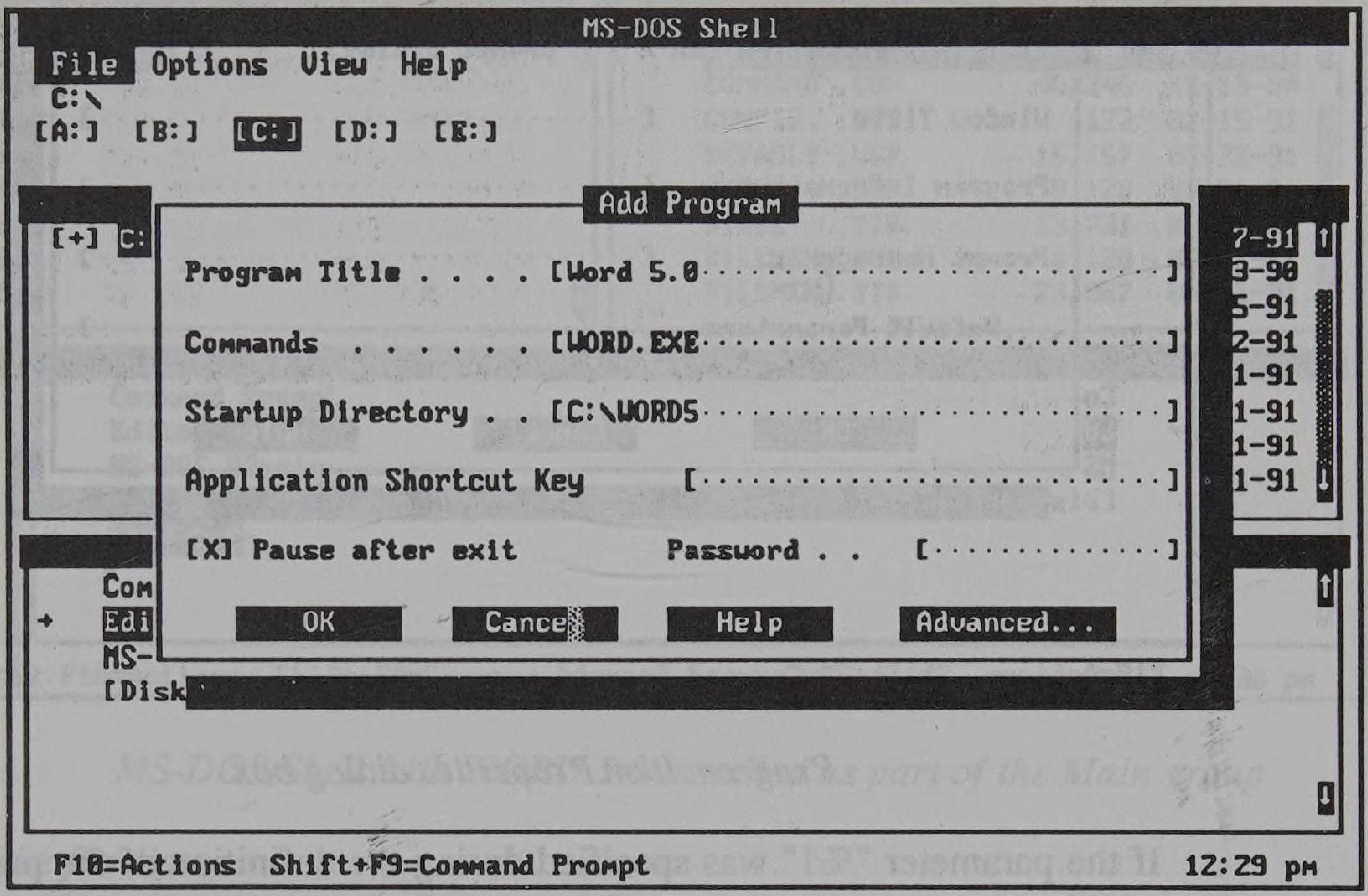
A dialog box is displayed for defining a new program object. Select the "Program Item" option as the new object.


MS-DOS Shell
File
Options
View
Help
C:\
[A:] [B:] [C:] [D:] [E:]
[+] C:\
New Program Object
New
( ) Program Group
(•) Program Item
OK
Cancel
C:\*.*
Main
Command Prompt
Editor
MS-DOS QBasic
[Disk Utilities]
F10=Actions Shift+F9=Command Prompt
12:28 pm

```
   306  02-27-91
46,246  12-13-90
   122  02-15-91
15,157  02-22-91
 8,128  03-01-91
23,731  03-01-91
 8,128  03-01-91
23,967  03-01-91
```

Selecting Program Item from New Program Object

Another dialog box is displayed in which you can enter the appropriate information about the file. You'll need to enter the program item to be displayed in the Main group, the command needed to start the program, the start up directory, any application shortcut key, which can be used as a "hot key" to instantly switch to the program, an optional password for access to the program and other options.



Program Title . . . . [Word 5.0]
Commands . . . . . . [WORD.EXE]
Startup Directory . . [C:\WORD5]
Application Shortcut Key [ ]
[X] Pause after exit
Password . . [ ]

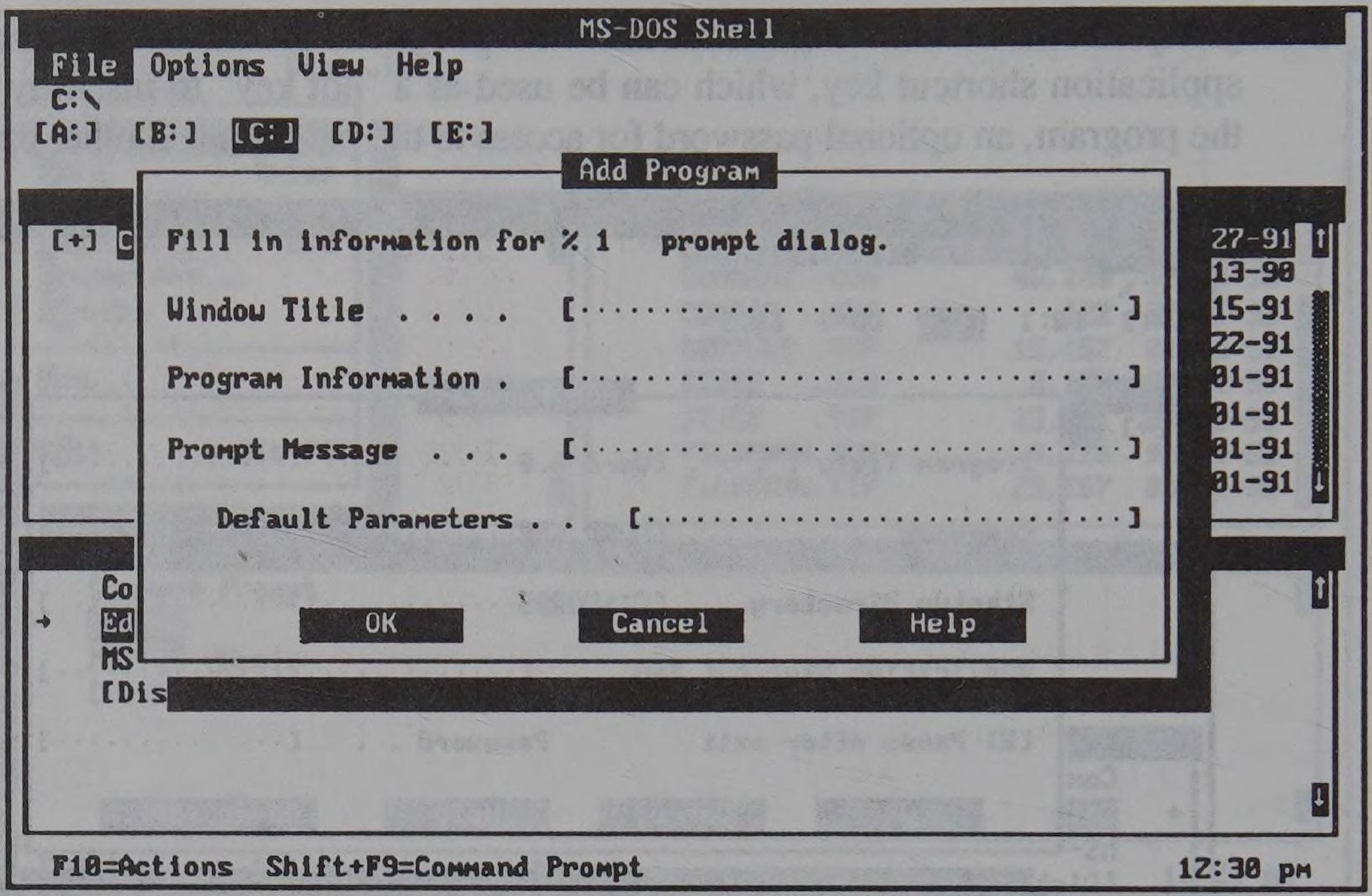
Adding a program to the Main group in the MS-DOS Shell

- The Program title text box corresponds to the name of the program as it should appear in the program list. In the above example, the program title is Word 5.0.

  The Commands text box is needed to start the program. This can be the name of the program or a batch file. To start Word in the C:\WORD directory, enter C:\WORD. Just as parameters can be specified in a batch file, parameters can be passed to start a program and load a document. The command in the preceding example could be written as:

  ```
  C:\WORD\WORD %1
  ```

  Press Enter or click on "OK" with the mouse to complete the definition of the program. Next you'll be prompted to supply information that will be used when the program is called.


MS-DOS Shell
File
Options
View
Help
C:\
[A:]
[B:]
[C:]
[D:]
[E:]
Add Program
[+]
27-91
13-90
15-91
22-91
01-91
01-91
01-91
01-91
Window Title . . . .
Program Information .
Prompt Message . . .
Default Parameters . .
Co
Ed
MS
[Dis
OK
Cancel
Help
F10=Actions Shift+F9=Command Prompt
12:30 pm

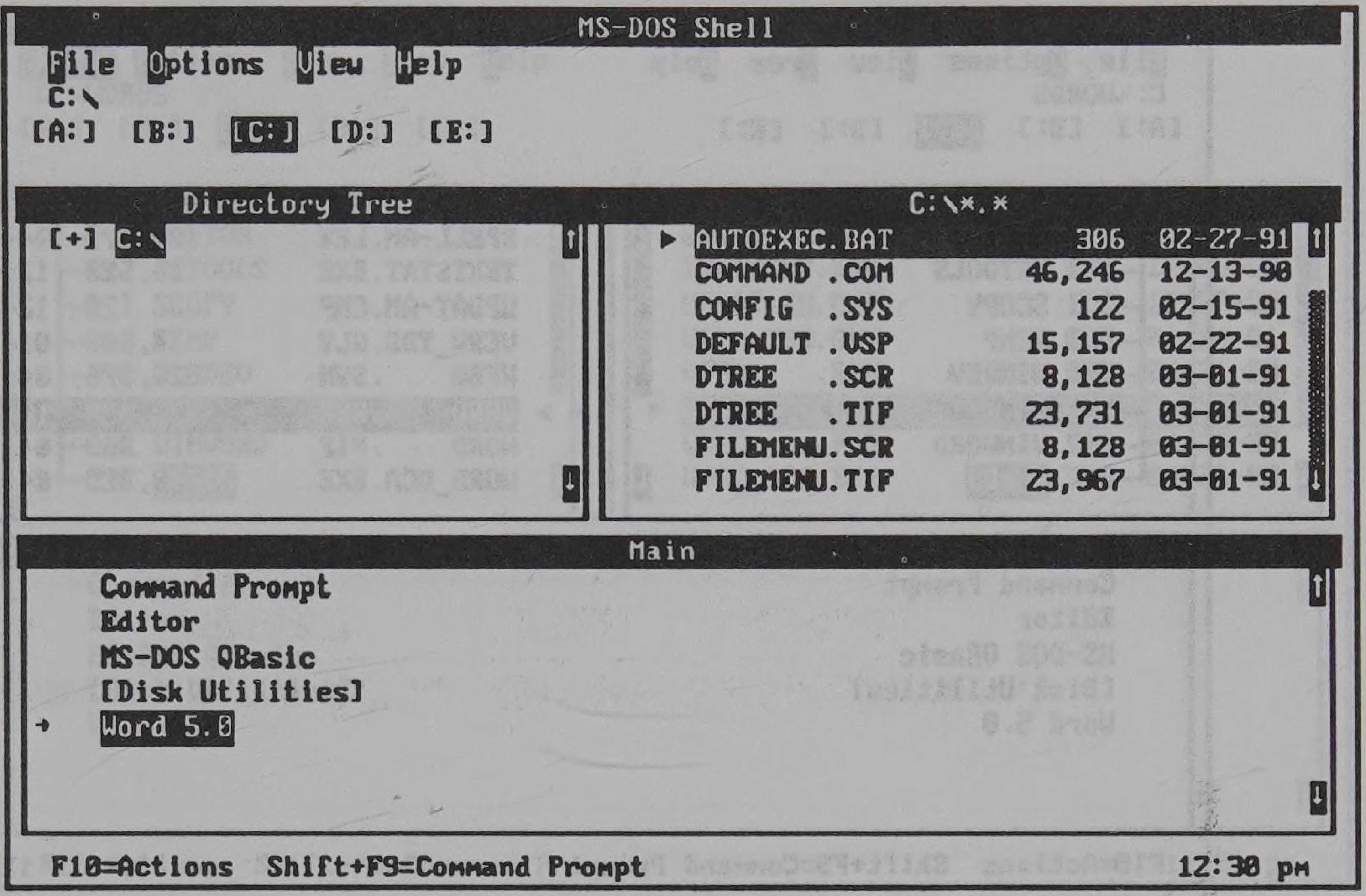
Fill in information for % 1 prompt dialog.

Program Item Properties dialog box

If the parameter "%1" was specified during the definition of the program, an additional screen appears for defining specific parameters prior to executing the program.

Enter the required information or press Enter to ignore this prompt.

After the program has been installed on the program list as described above, it can be started by selecting it from the Main group with either the mouse or the cursor keys.



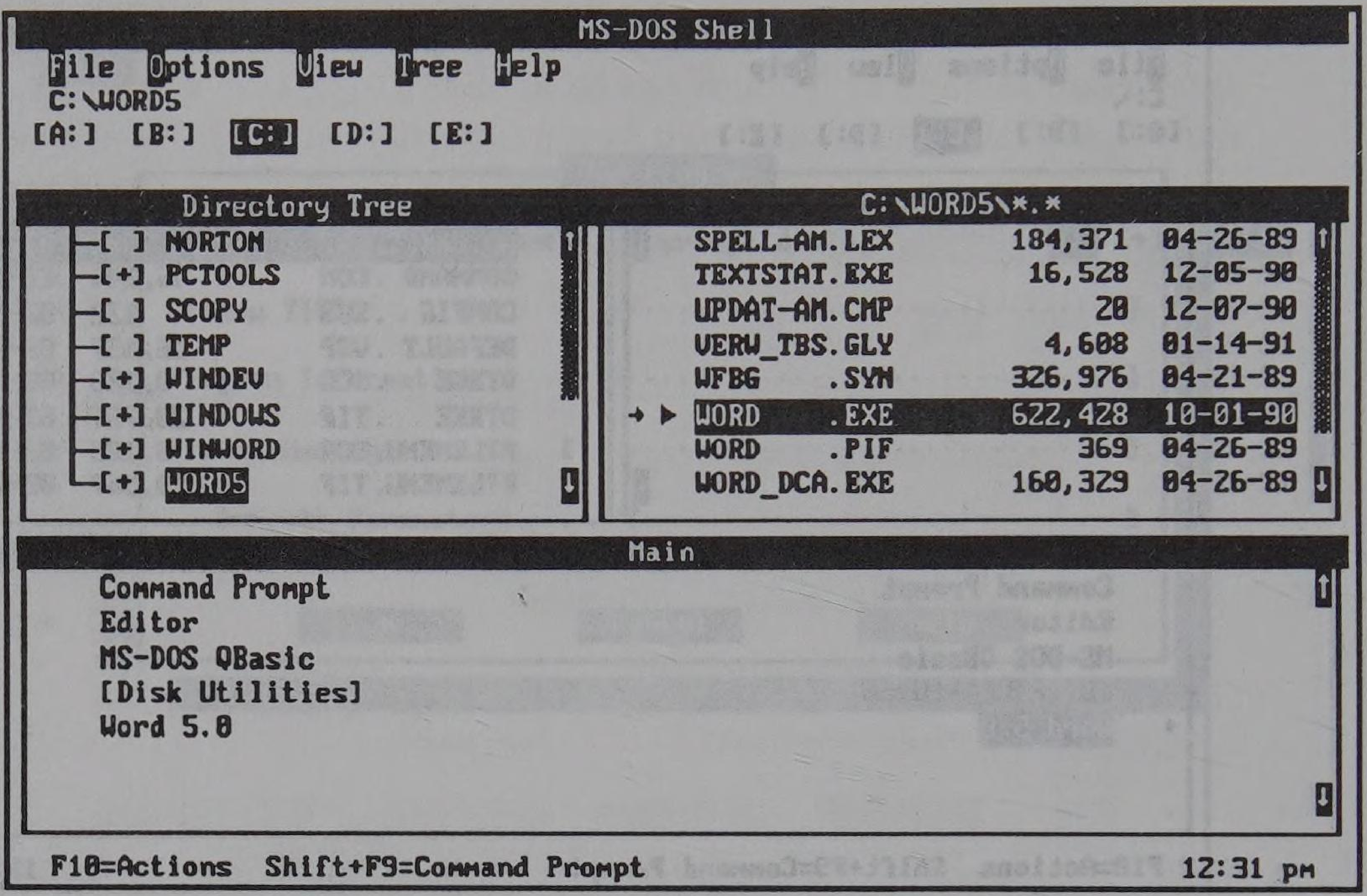
```
                              MS-DOS Shell
 File  Options  View  Help
 C:\
[A:]  [B:]  [C:]  [D:]  [E:]

          Directory Tree                               C:\*.*
 [+] C:\                           ► AUTOEXEC.BAT           306  02-27-91
                                     COMMAND  .COM       46,246  12-13-90
                                     CONFIG   .SYS          122  02-15-91
                                     DEFAULT  .VSP       15,157  02-22-91
                                     DTREE    .SCR        8,128  03-01-91
                                     DTREE    .TIF       23,731  03-01-91
                                     FILEMENU.SCR         8,128  03-01-91
                                     FILEMENU.TIF        23,967  03-01-91

                                  Main
      Command Prompt
      Editor
      MS-DOS QBasic
      [Disk Utilities]
 →    Word 5.0

 F10=Actions  Shift+F9=Command Prompt                          12:30 pm
```

MS-DOS Shell with WORD 5.0 installed as part of the Main group

The file list

Any executable program can be started from the file list, which is located on the right side of the screen. Locate the program by selecting the drive, directory and any subdirectory in the appropriate Shell areas, then, in the right area, select the program file to start.

This program file must have either a .COM, .EXE or .BAT extension. To start the program file, press Enter or use the mouse to double-click on the filename.



```
                                 MS-DOS Shell
File  Options  View  Tree  Help
C:\WORDS
[A:]  [B:]  [C:]  [D:]  [E:]

            Directory Tree                              C:\WORDS\*.*
  ─[ ] NORTON                           SPELL-AM.LEX        184,371  04-26-89
  ─[+] PCTOOLS                          TEXTSTAT.EXE         16,528  12-05-90
  ─[ ] SCOPY                            UPDAT-AM.CMP             20  12-07-90
  ─[ ] TEMP                             VERW_TBS.GLY          4,608  01-14-91
  ─[+] WINDEV                           WFBG     .SYN       326,976  04-21-89
  ─[+] WINDOWS                      → ► WORD     .EXE       622,428  10-01-90
  ─[+] WINWORD                          WORD     .PIF           369  04-26-89
  └[+] WORDS                            WORD_DCA.EXE        160,329  04-26-89

                                     Main
      Command Prompt
      Editor
      MS-DOS QBasic
      [Disk Utilities]
      Word 5.0

 F10=Actions   Shift+F9=Command Prompt                                12:31 pm
```

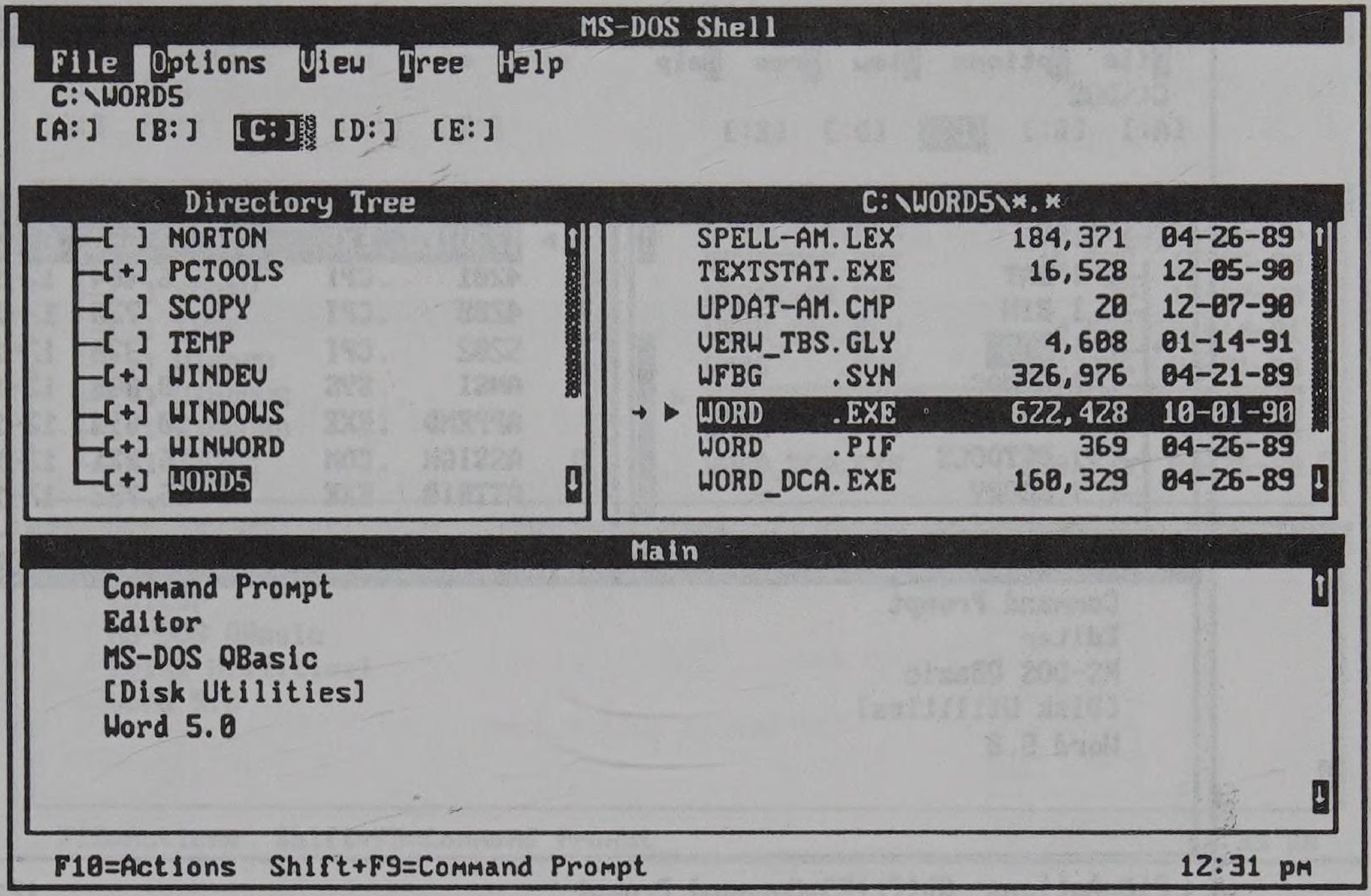
Preparing to start WORD.EXE from the MS-DOS Shell file list

Don't start TSR (Terminate and Stay Resident) programs from the MS-DOS Shell.

The following section explains the steps for starting a program from the file list in greater detail.

Selecting a drive

Use Tab to move the selection highlight in the MS-DOS Shell to the drive icons or use the mouse pointer to point to the letter of the required drive and press the left mouse button.



```
                               MS-DOS Shell
 File  Options  View  Tree  Help
  C:\WORDS5
 [A:]  [B:]  [C:]  [D:]  [E:]

          Directory Tree                          C:\WORDS5\*.*
   ├─[ ] NORTON                         SPELL-AM.LEX        184,371  04-26-89
   ├─[+] PCTOOLS                        TEXTSTAT.EXE         16,528  12-05-90
   ├─[ ] SCOPY                          UPDAT-AM.CMP             20  12-07-90
   ├─[ ] TEMP                           VERW_TBS.GLY          4,608  01-14-91
   ├─[+] WINDEV                         WFBG      .SYN      326,976  04-21-89
   ├─[+] WINDOWS                    → ► WORD      .EXE      622,428  10-01-90
   ├─[+] WINWORD                        WORD      .PIF          369  04-26-89
   └─[+] WORDS5                         WORD_DCA.EXE        160,329  04-26-89

                                    Main
    Command Prompt
    Editor
    MS-DOS QBasic
    [Disk Utilities]
    Word 5.0

 F10=Actions  Shift+F9=Command Prompt                                  12:31 pm
```

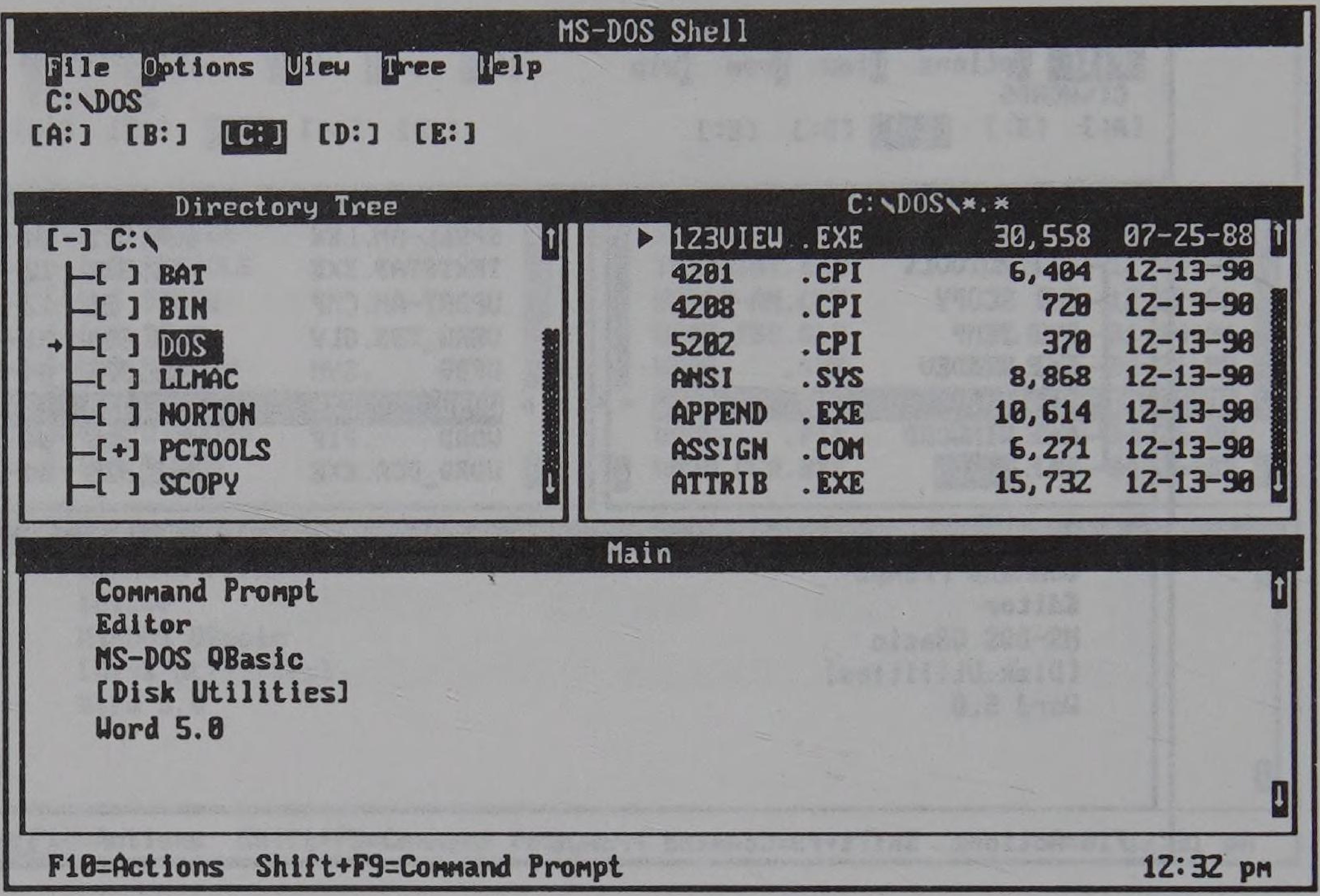
Positioning the cursor to select drive C:

Use the cursor keys to highlight the required drive. To activate this drive, press (Enter) or double-click on it with the mouse button.

The tree structure of the directories in the selected drive appears in the left area of the MS-DOS Shell, as shown above.

Selecting a directory

Use (Tab) to activate the left area or click on the Directory Tree area with the mouse. Using either the cursor keys or the mouse to highlight (select) the required directory. The names of files stored in this directory are displayed in alphabetical order in the right area.



```
                              MS-DOS Shell
 File  Options  View  Tree  Help
 C:\DOS
 [A:]  [B:]  [C:]  [D:]  [E:]

          Directory Tree                              C:\DOS\*.*
 [-] C:\                                ▶ 123VIEW .EXE       30,558  07-25-88
   ├─[ ] BAT                              4201    .CPI        6,404  12-13-90
   ├─[ ] BIN                              4208    .CPI          720  12-13-90
  →├─[ ] DOS                              5202    .CPI          370  12-13-90
   ├─[ ] LLMAC                            ANSI    .SYS        8,868  12-13-90
   ├─[ ] NORTON                           APPEND  .EXE       10,614  12-13-90
   ├─[+] PCTOOLS                          ASSIGN  .COM        6,271  12-13-90
   ├─[ ] SCOPY                            ATTRIB  .EXE       15,732  12-13-90

                                    Main
        Command Prompt
        Editor
        MS-DOS QBasic
        [Disk Utilities]
        Word 5.0

 F10=Actions  Shift+F9=Command Prompt                               12:32 pm
```

Note the arrow displayed next to the DOS directory in the Directory Tree and DOS files in the file list (right area)

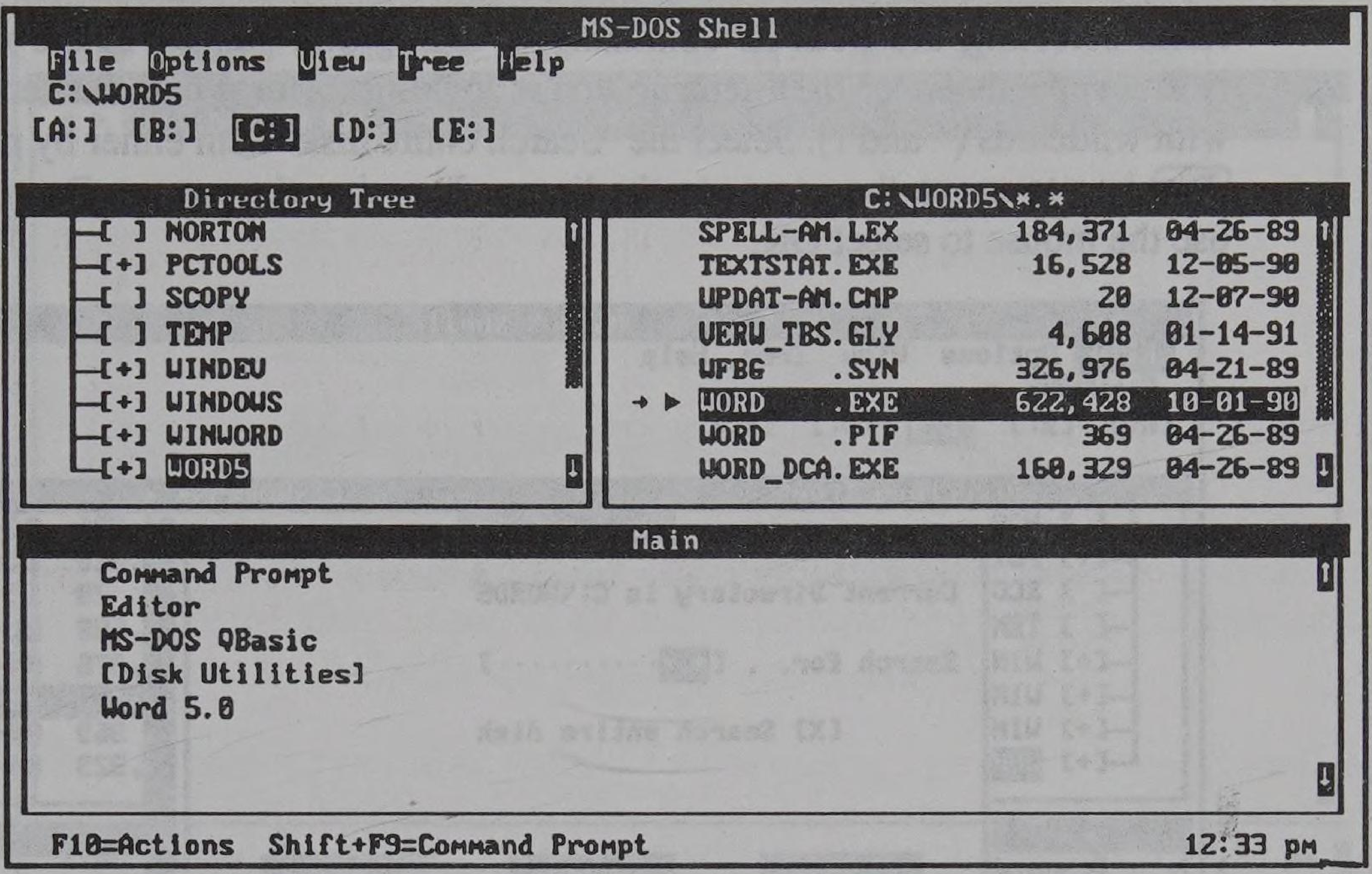
Selecting (marking) a file and starting it

Use Tab to switch to the right area or select it with the mouse pointer. Select the program file to run (COM, EXE, BAT) by using the ↑ and ↓ cursor keys to move by line, the PgUp and PgDn keys to move by page or by pointing with the mouse pointer. Pressing Home moves to the top of the file list. Pressing End moves to the end of the file list.

If you're using the mouse, you can click both direction arrows on the scroll bar to scroll the contents of the file list. Clicking in the grey field enables you to jump to the next screen within the file list.

You can select the file to be run by using the ↑ and ↓ cursor keys and stopping at the desired file. The filename appears as inverse text (highlighted), which indicates that the file is marked. The program can be started by pressing Enter.

If you're using a mouse, select a file by clicking the left mouse button. Double-click the left mouse button to start the program.

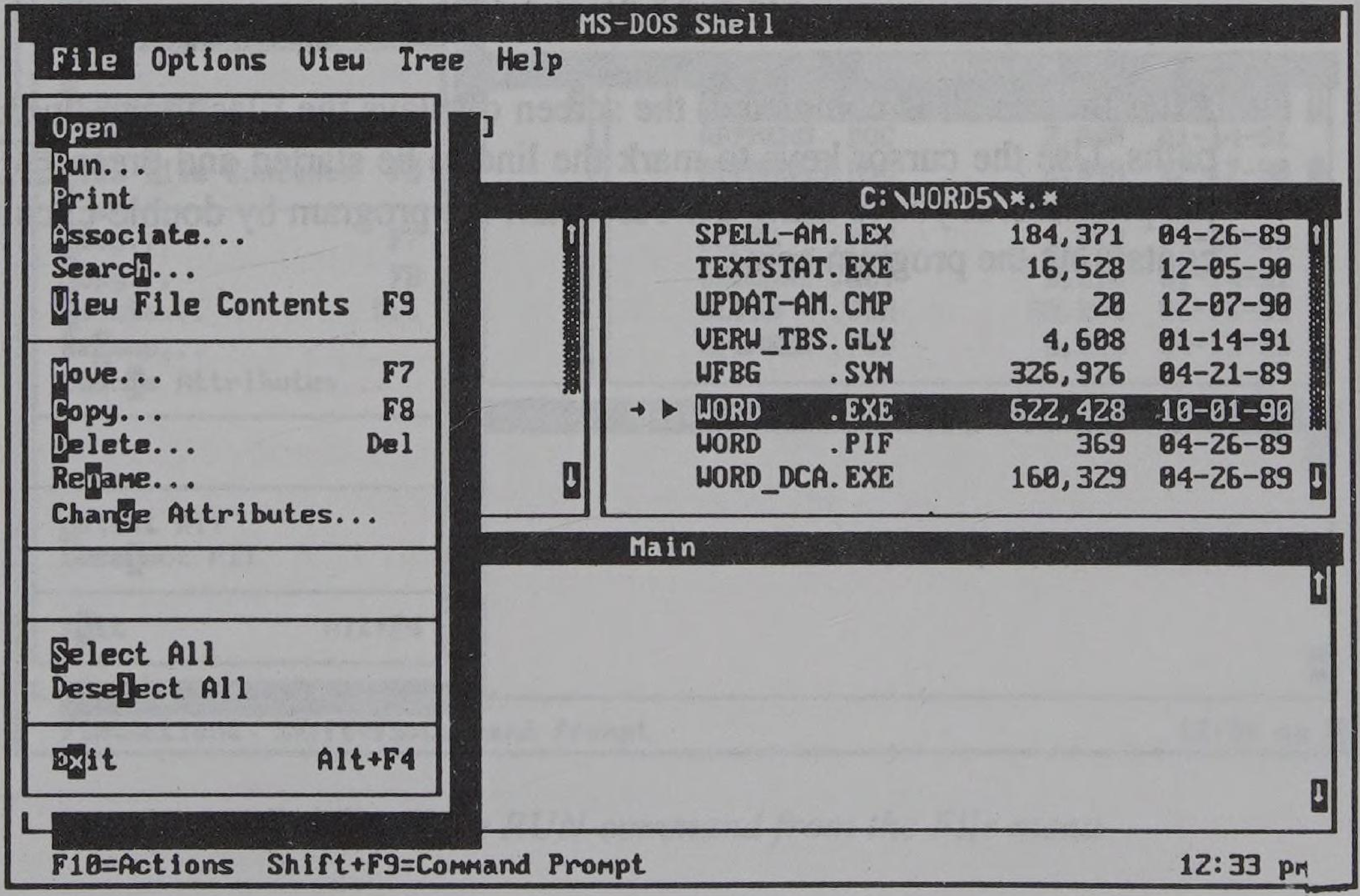


```
                                MS-DOS Shell
 File  Options  View  Tree  Help
 C:\WORD5
 [A:]  [B:]  [C:]  [D:]  [E:]
          Directory Tree                          C:\WORD5\*.*
   ├[ ] NORTON                          SPELL-AM.LEX   184,371  04-26-89
   ├[+] PCTOOLS                         TEXTSTAT.EXE    16,528  12-05-90
   ├[ ] SCOPY                           UPDAT-AM.CMP        20  12-07-90
   ├[ ] TEMP                            VERW_TBS.GLY     4,608  01-14-91
   ├[+] WINDEV                          WFBG    .SYN   326,976  04-21-89
   ├[+] WINDOWS                     → ► WORD    .EXE   622,428  10-01-90
   ├[+] WINWORD                         WORD    .PIF       369  04-26-89
   └[+] WORD5                           WORD_DCA.EXE   160,329  04-26-89
                                     Main
      Command Prompt
      Editor
      MS-DOS QBasic
      [Disk Utilities]
      Word 5.0

 F10=Actions  Shift+F9=Command Prompt                                12:33 pm
```

WORD.EXE in the file list is selected; pressing Enter *will run this program*

Searching for a file and starting it

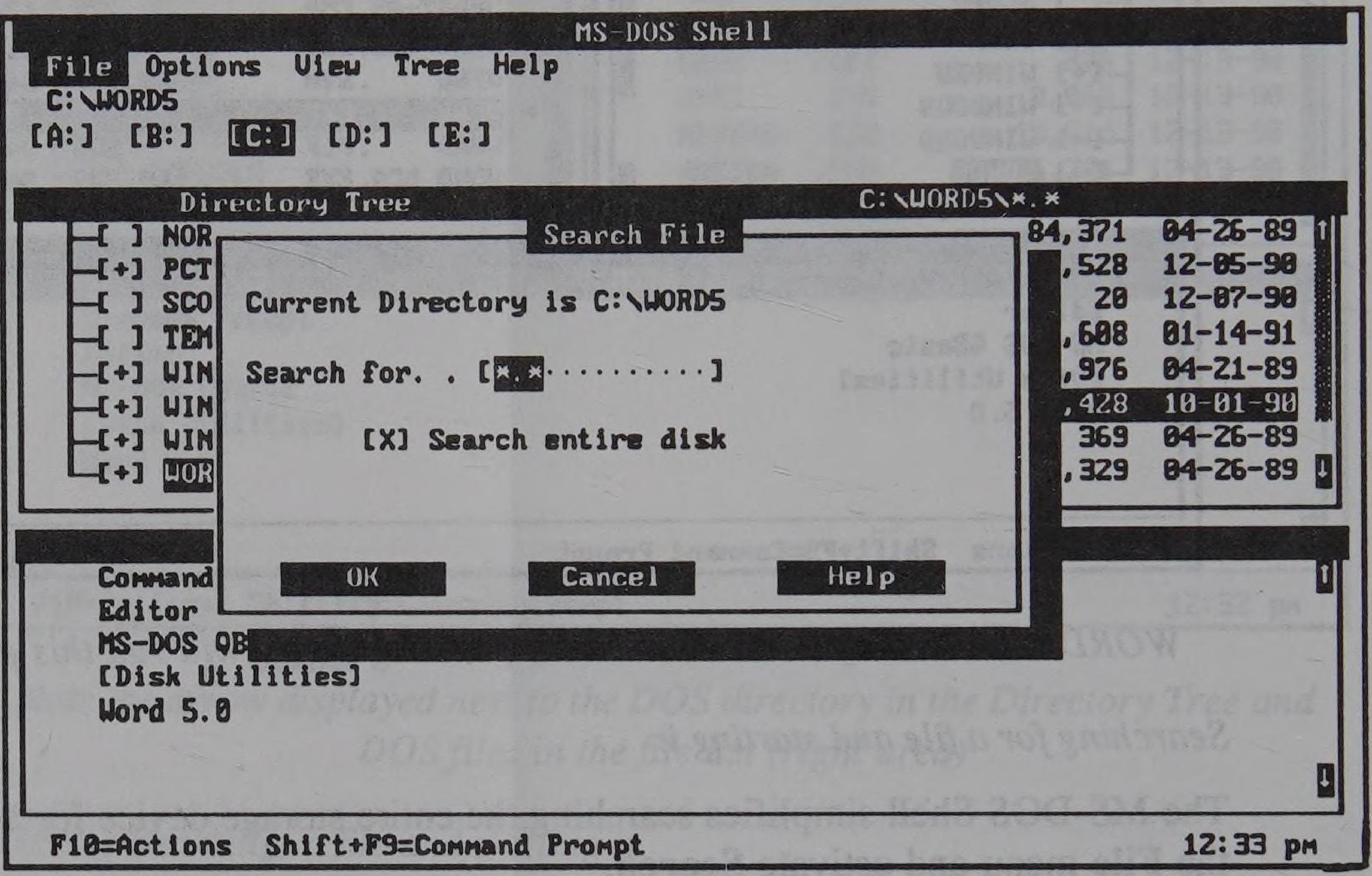
The MS-DOS Shell simplifies searching the entire storage device for a file. Open the **File** menu and activate **Search**.



```
                                MS-DOS Shell
 File  Options  View  Tree  Help
 ┌──────────────────────────┐
 │ Open                     │  ]
 │ Run...                   │
 │ Print                    │                C:\WORD5\*.*
 │ Associate...             │       SPELL-AM.LEX   184,371  04-26-89
 │ Search...                │       TEXTSTAT.EXE    16,528  12-05-90
 │ View File Contents   F9  │       UPDAT-AM.CMP        20  12-07-90
 ├──────────────────────────┤       VERW_TBS.GLY     4,608  01-14-91
 │ Move...              F7  │       WFBG    .SYN   326,976  04-21-89
 │ Copy...              F8  │   → ► WORD    .EXE   622,428  10-01-90
 │ Delete...            Del │       WORD    .PIF       369  04-26-89
 │ Rename...                │       WORD_DCA.EXE   160,329  04-26-89
 │ Change Attributes...     │
 ├──────────────────────────┤           Main
 │                          │
 ├──────────────────────────┤
 │ Select All               │
 │ Deselect All             │
 ├──────────────────────────┤
 │ Exit             Alt+F4  │
 └──────────────────────────┘
 F10=Actions  Shift+F9=Command Prompt                                12:33 pm
```

The File menu in MS-DOS Shell

After selecting the **Search** command, a dialog box similar to the one shown below appears. Enter the filename you're looking for or a partial filename along with wildcards (* and ?). Select the "Search entire disk" item either by pressing the Tab key to move the cursor to the line or by using the mouse. Press Enter or use the mouse to select OK.

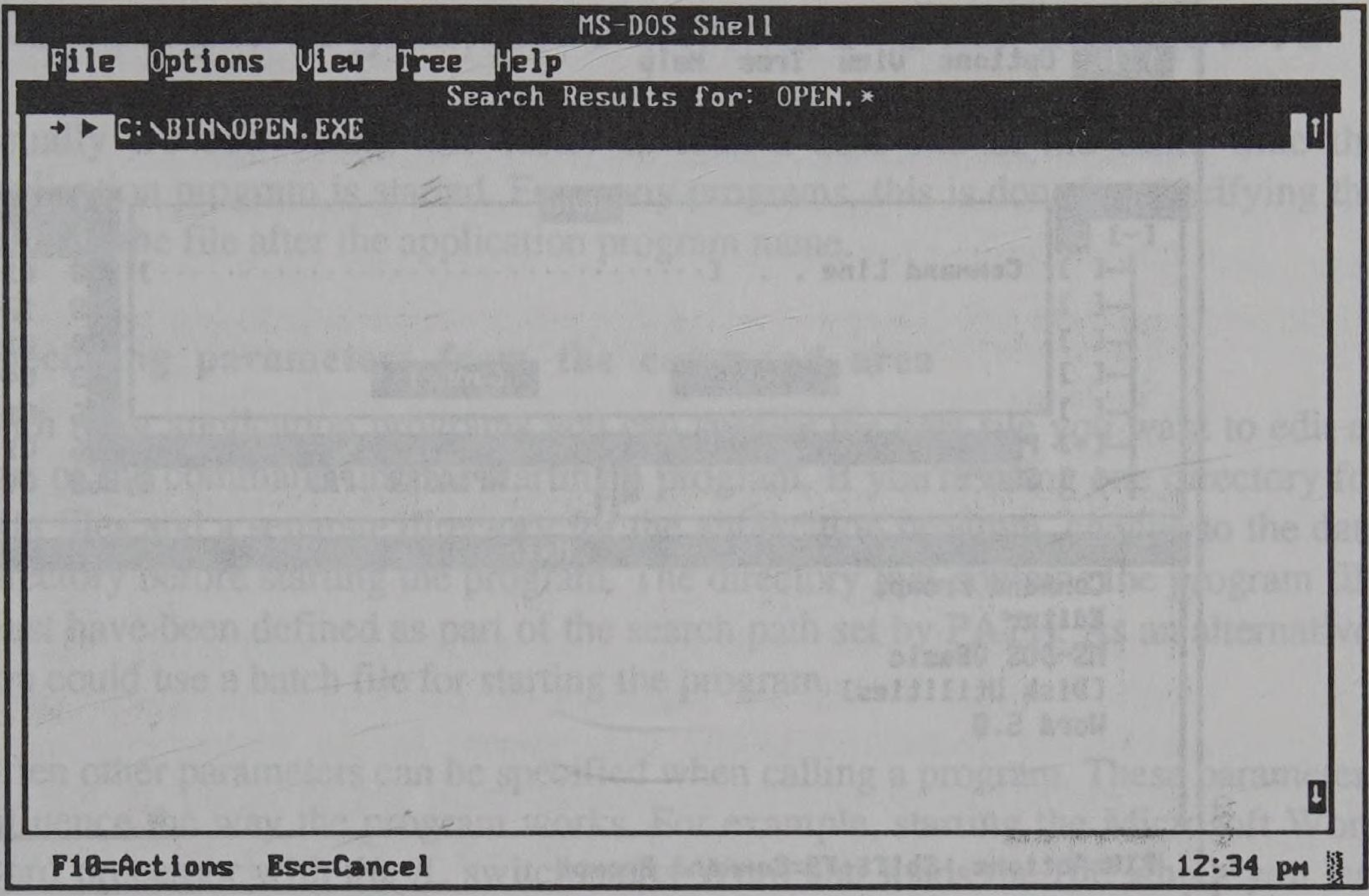

MS-DOS Shell
File Options View Tree Help
C:\WORDS
[A:] [B:] [C:] [D:] [E:]
Directory Tree
C:\WORDS\*.*
Search File
OK
Cancel
Help
F10=Actions Shift+F9=Command Prompt
12:33 pm

```
Current Directory is C:\WORDS

Search for. . [*.*..........]

      [X] Search entire disk
```

```
[ ] NOR     84,371  04-26-89
[+] PCT       ,528  12-05-90
[ ] SCO         20  12-07-90
[ ] TEM       ,608  01-14-91
[+] WIN       ,976  04-21-89
[+] WIN       ,428  10-01-90
[+] WIN        369  04-26-89
[+] WOR       ,329  04-26-89

Command
Editor
MS-DOS QB
[Disk Utilities]
Word 5.0
```

MS-DOS Shell Search dialog box

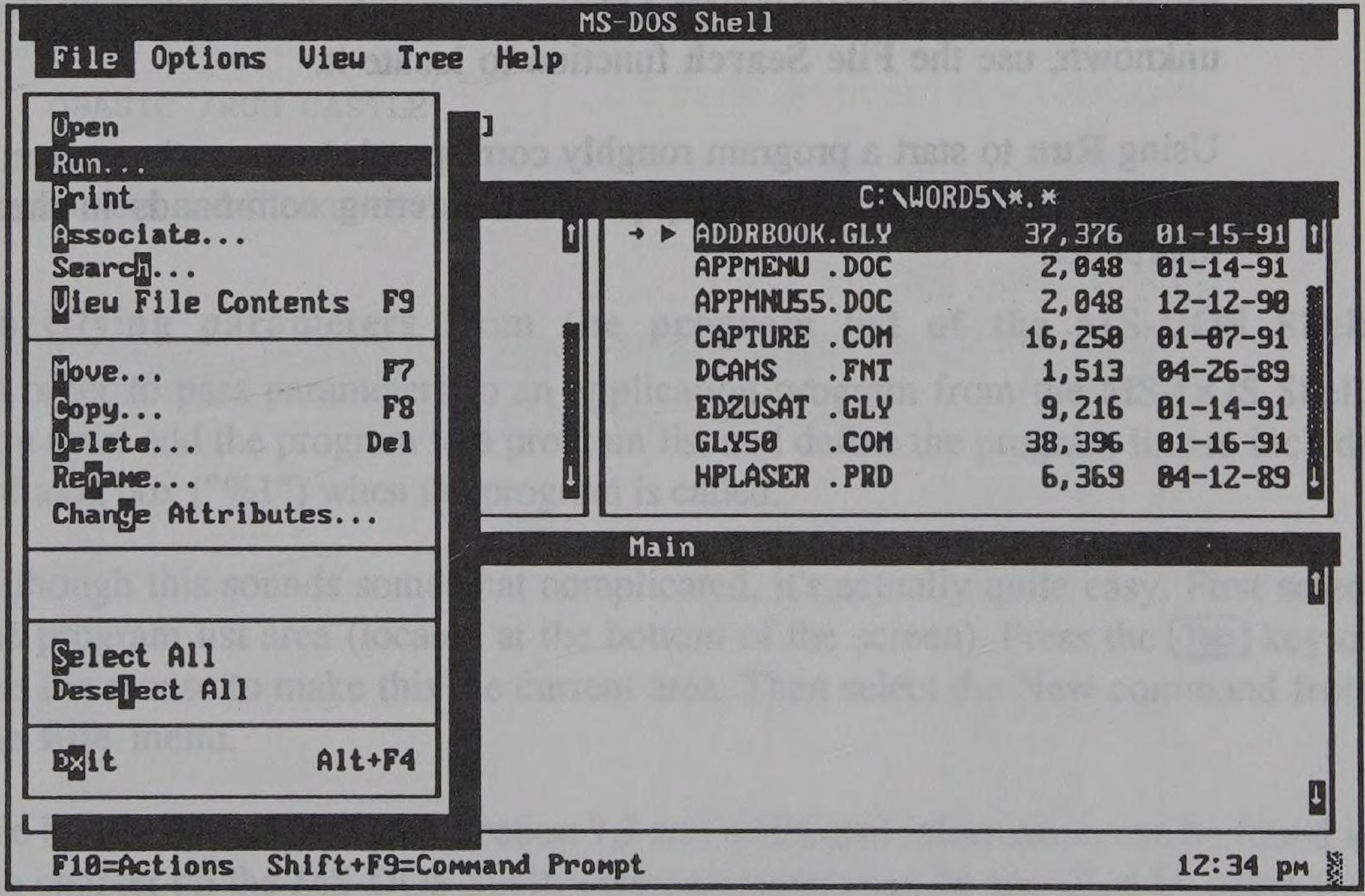
After the search is completed, the screen displays the files found, including their paths. Use the cursor keys to mark the line to be started and press Enter to start the program. If you're using a mouse, start the program by double-clicking the line containing the program name.


MS-DOS Shell
File Options View Tree Help
Search Results for: OPEN.*
F10=Actions Esc=Cancel
12:34 pm

```
→ ▶ C:\BIN\OPEN.EXE
```

*The results of searching for OPEN.**

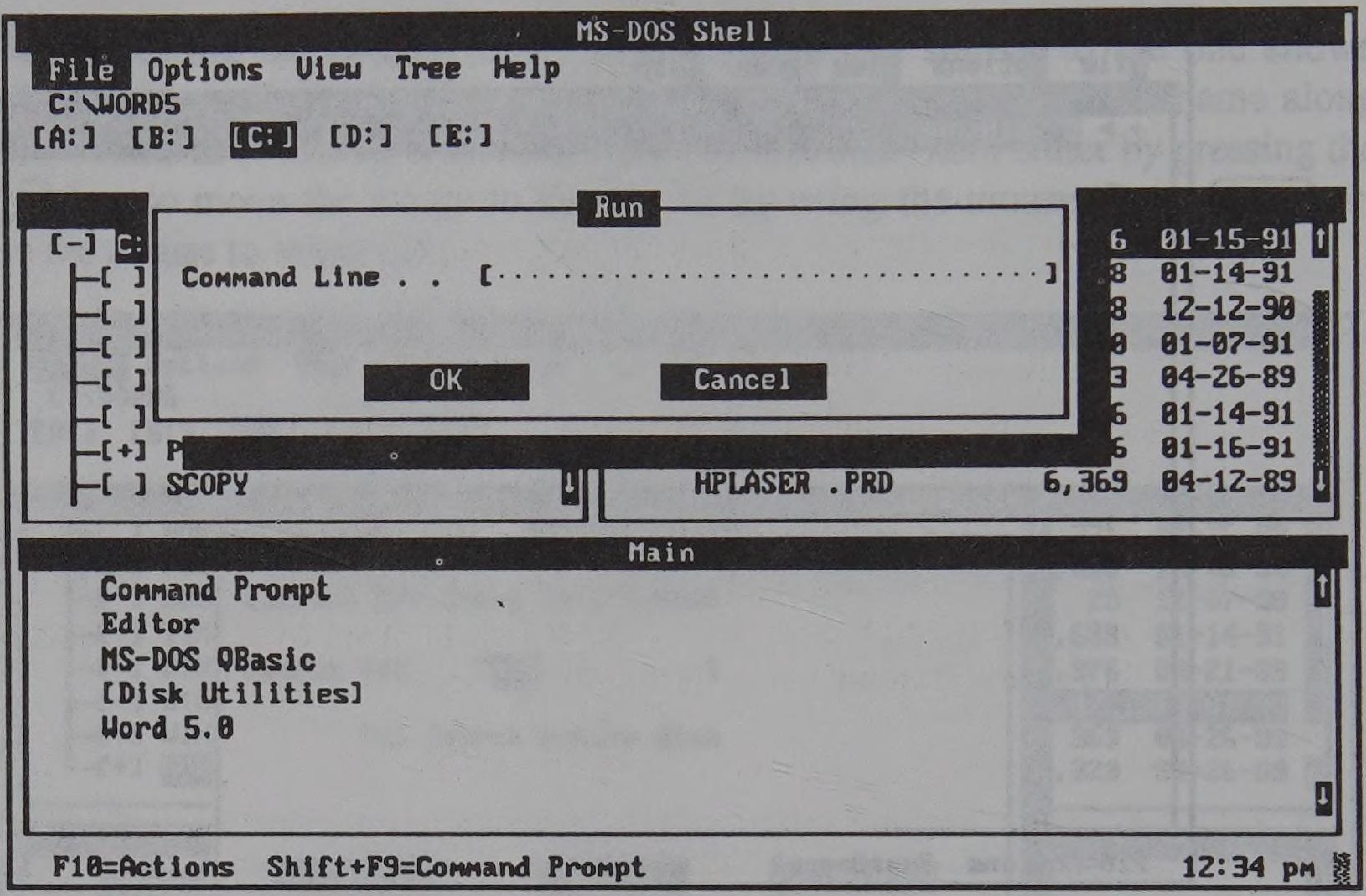
Using RUN to start a program


MS-DOS Shell
File Options View Tree Help
Open
Run...
Print
Associate...
Search...
View File Contents F9
Move... F7
Copy... F8
Delete... Del
Rename...
Change Attributes...
Select All
Deselect All
Exit Alt+F4
Main
F10=Actions Shift+F9=Command Prompt
12:34 pm

```
C:\WORD5\*.*
→ ▶ ADDRBOOK.GLY    37,376  01-15-91
    APPMENU .DOC     2,048  01-14-91
    APPMNU55.DOC     2,048  12-12-90
    CAPTURE .COM    16,250  01-07-91
    DCAMS   .FNT     1,513  04-26-89
    EDZUSAT .GLY     9,216  01-14-91
    GLY50   .COM    38,396  01-16-91
    HPLASER .PRD     6,369  04-12-89
```

Selecting the RUN command from the File menu

The third way to start a program from the MS-DOS Shell is by using the **Run...** command from the **File** menu. When this command is activated, a dialog box appears which allows you to enter the name of the program you want to start.


MS-DOS Shell
File Options View Tree Help
C:\WORDS
[A:] [B:] [C:] [D:] [E:]
Run
Command Line . . [..............................]
OK
Cancel
SCOPY
HPLASER .PRD 6,369 04-12-89
Main
Command Prompt
Editor
MS-DOS QBasic
[Disk Utilities]
Word 5.0
F10=Actions Shift+F9=Command Prompt
12:34 pm

The Run command dialog box

If the program is in a directory that has been defined by PATH, simply enter the program name and press Enter. Otherwise, specify the entire path. If the path is unknown, use the **File Search** function to locate it.

Using **Run** to start a program roughly corresponds to executing a command line. The same basic conditions apply as for entering commands in the command interpreter.

7.4 Starting Programs With Files Or Parameters

Usually it's convenient and faster to load a data file at the same time the application program is started. For many programs, this is done by specifying the name of the file after the application program name.

Specifying parameters from the command area

With most application programs you can specify the data file you want to edit as part of the command line for starting a program. If you're using one directory for data files and a separate directory for the application program, change to the data directory before starting the program. The directory that contains the program file must have been defined as part of the search path set by PATH. As an alternative, you could use a batch file for starting the program.

Often other parameters can be specified when calling a program. These parameters influence the way the program works. For example, starting the Microsoft Word word processor with the /L switch starts Word and loads the file which you last edited.

The /RUN switch in QBasic, which is part of MS-DOS 5.0, runs a QBasic program without having to access the QBasic menus to load and run the QBasic program. Specify the name of the program in the command line:

```
QBASIC /RUN CASTLE
```

Parameters and filenames can also be used with the MS-DOS Shell.

Specifying parameters from the program list of the MS-DOS Shell

In order to pass parameters to an application program from the MS-DOS Shell, you must add the program to a program list and define the program line to include a dialog box ("%1") when the program is called.

Although this sounds somewhat complicated, it's actually quite easy. First select the program list area (located at the bottom of the screen). Press the Tab key or use the mouse to make this the current area. Then select the **New** command from the **File** menu.

We briefly discussed this in Section 7.3 and additional information can be found in the chapter on the MS-DOS Shell. Two parameters can be specified by using two dialog boxes ("%1" and "%2") after the name of the command.

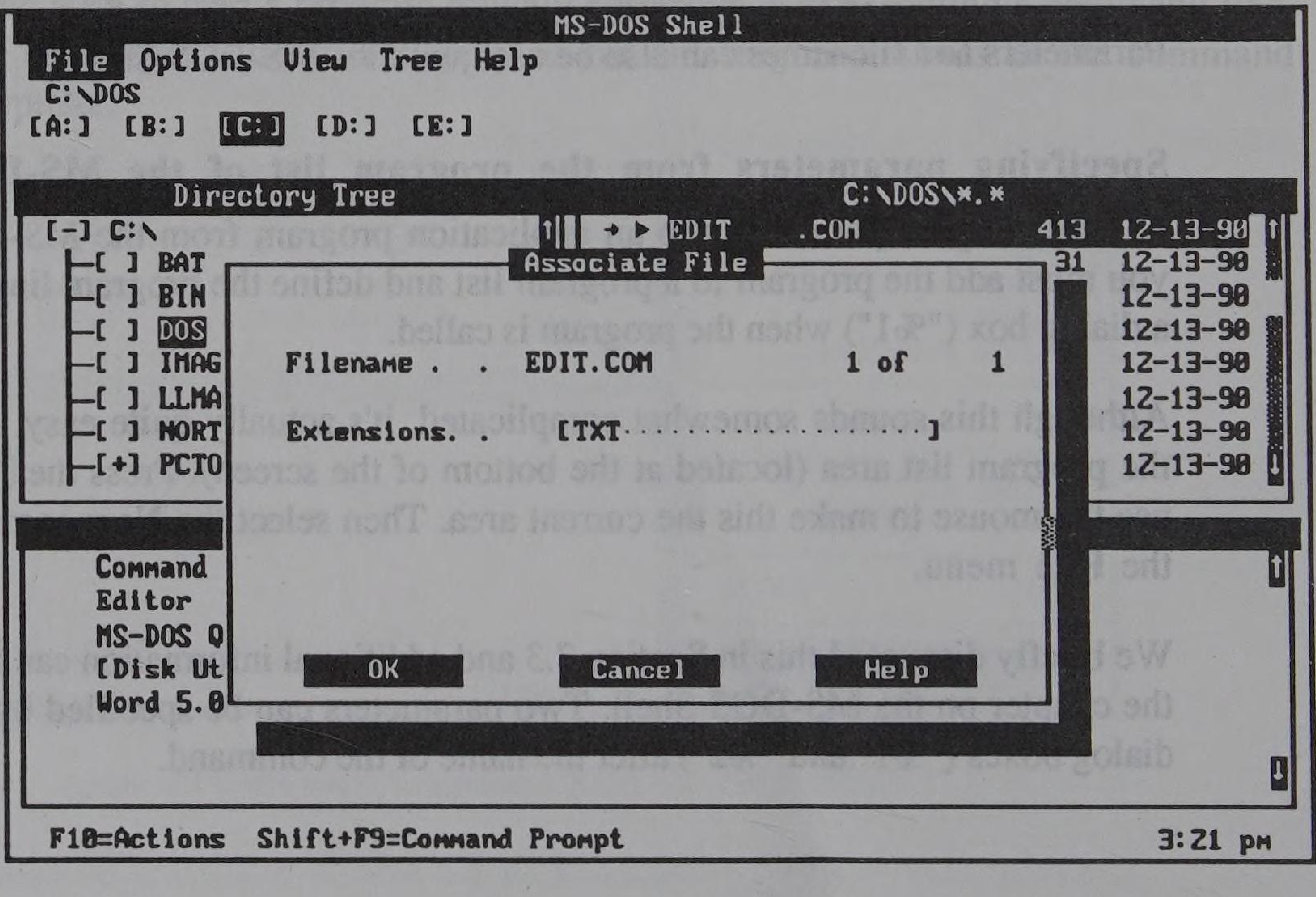
Opening files and programs from the file list

Although additional parameters can't be specified when starting a program from the MS-DOS Shell file list, an interesting and useful option is available when you open files with this list. File extensions can be defined as being "associated" with a specific application program.

When you associate files and an application, you are specifying that any file with a particular extension is to be used with a specific application. For example, you can associate all files having the .TXT extension with the MS-DOS EDIT command. Then when you select a TXT file and open it, the MS-DOS Editor is used to edit that file.

To associate all files with the .TXT extension with EDIT, select the DOS directory in the file list and highlight the EDIT.COM program. Then select the **Associate** command in the **File** menu. In the dialog box, enter the file extensions that should be associated with the program in the line labeled "Extensions". In this example, type TXT. Pressing Enter completes the association. Note that multiple file extensions can be associated with a single program.

Even though twenty extensions can be associated with a program file, a specific extension can be associated with only one program. For example, .TXT can be associated with the MS-DOS Editor (the EDIT command) but it can't also be associated with Microsoft Word 5.0.


MS-DOS Shell
File
Options
View
Tree
Help
C:\DOS
[A:] [B:] [C:] [D:] [E:]
Directory Tree
C:\DOS\*.*
[-] C:\
[ ] BAT
[ ] BIN
[ ] DOS
[ ] IMAG
[ ] LLMA
[ ] NORT
[+] PCTO
EDIT .COM 413 12-13-90
31 12-13-90
12-13-90
Associate File
Command
Editor
MS-DOS Q
[Disk Ut
Word 5.0
OK
Cancel
Help
F10=Actions Shift+F9=Command Prompt
3:21 pm

Filename . . EDIT.COM 1 of 1

Extensions. . [TXT.....................]

The Associate File dialog box

Now when you activate a file, with the .TXT extension, in the file list by pressing Enter or by double-clicking the mouse button, EDIT is automatically called with the file so that you can edit the file.

You could also associate the .BAT extension with the EDIT program. This would enable you to make changes in the AUTOEXEC.BAT or any batch file, very easily. Simply double-click on the batch file in the file list and it automatically loads into EDIT.

7.5 Quitting Programs

Most programs have a special command for exiting the program. Refer to the program's user manual for specific instructions. If the application program requires a large amount of memory, a part of the command interpreter is removed from the memory. The command interpreter must later be reloaded into memory when you exit from such a program.

Never switch off the computer without first exiting the application program properly. If you just switch off the computer, you may lose some of your data. If there is a power failure while using a program or if the computer seems to be dead (no response to the keyboard, etc.), restart the computer and use CHKDSK /F to check the hard drive or diskette for any possible problems.

If you're using MS-DOS from a hard drive and you're still prompted to insert a diskette with COMMAND.COM when you exit an application program, then you can use the following command that tells DOS where on the hard drive to find the command interpreter. If the COMMAND.COM file is in the root directory, the command line looks like the following:

```
SET COMSPEC=C:\COMMAND.COM
```

If the command interpreter is in a subdirectory of the hard drive, enter the complete path. Of course, you can insert this command into your AUTOEXEC.BAT file.

Quitting the MS-DOS Shell

Never exit the MS-DOS Shell by simply switching off your computer. Quit the MS-DOS Shell by pressing Alt + F4 or F3 to ensure that there are no more programs in the list of active programs (the Task Switcher, which will be discussed later) and all of the settings of the MS-DOS Shell have been saved to the MS-DOS Shell INI file.

The MS-DOS Shell INI file (DOSSHELL.INI) contains information about changes and preferences you've selected for the MS-DOS Shell. Simply switching off your computer won't save any of these settings. So this information will be lost.

7.6 Avoiding Problems With Programs

By taking simple precautions, you can avoid potential problems with application programs. The following are the most important features and precautions to keep in mind.

Backing up data

Plan for safe guarding your data while you're installing a new application program on your hard drive. Install the program in it's own subdirectory and set up another subdirectory for storing the data created with the application program. After it's installed, it's not as easy to change the way in which the data is stored.

Setting the time and the date

If your computer doesn't have a real time clock, add the MS-DOS DATE and TIME commands to your AUTOEXEC.BAT file and enter the current values when starting your computer. The date and time stamps will help you determine which version of a file is most recent. Users frequently create several versions of a basic document and then aren't able to tell them apart. So the date and time of a file stored in the directory can be very helpful.

As we previously explained, you can also use the XCOPY, BACKUP and RESTORE commands to back up and restore data created before or after a certain date. This makes protecting your data easier because you can be much more selective about which files are backed up.

Many users don't back up their data because it seems inconvenient and tedious to spend time doing a back up. However, by storing your data in specific subdirectories, and following a few simple procedures, the amount of time spent backing up data can be reduced to a few minutes.

Separating programs and data

Create directories for application programs and their associated data. This saves time and energy when backing up data. Usually an installed program only needs to be backed up once because the program can be reinstalled from the original diskettes. However, since the data that's created is often changed, it needs to be backed up at regular intervals.

You should also divide data among several directories and ensure that a directory doesn't contain more data than will fit on a single diskette. This enables you to perform backups quickly and easily. Simply use the COPY command to back up data from each directory onto a diskette.

Choosing filenames

If the application program doesn't automatically assign filenames or extensions (.DOC for document files from a word processor), be sure that you choose filenames that match. This will enable you to use COPY or XCOPY with wildcards in order to copy all of the data that belong together.

Example 1

Suppose that you have spent time creating illustrations, tables and texts for a report. The files all have different filename extensions since each application program creates a different file type. Create a separate directory for all of the files associated with the report so that all of the data can be backed up easily.

If for some reason this can't be done, start each filename with the same characters (e.g., REP for report). Then all of the files on the entire hard drive that are associated with the report can be backed up to diskettes with a simple command:

```
BACKUP C:\REP*.* A: /S
```

Example 2

If you work primarily with a word processor and all of your text files have the .DOC extension, you can back them up like this:

- If the files will all fit on a diskette, type:

  ```
  XCOPY *.DOC A: /S
  ```

- To backup all of the text files on the hard drive, even if all of the files won't fit on a single diskette, use the following command:

  ```
  BACKUP C:\*.TXT A: /S
  ```

The /S switch specifies that the command applies to all of the subdirectories.

If you've created so much data for a project that backing up the entire directory is too complicated (requiring several diskettes) and the data that has been changed doesn't exceed the capacity of a single diskette, create a special batch file that only backs up the data that has been changed. Place this special PCX-DOC.BAT batch file in the same directory as the data.

For example, the PCX-DOC.BAT file could copy all files with the .DOC extension and all illustrations with the .PCX extension onto a diskette:

```
@ECHO OFF
ECHO Back up data to a diskette
ECHO Please insert diskette and press any key
PAUSE
COPY *.DOC A:
```

```
COPY *.PCX A:
ECHO Data backed up
ECHO ON
```

If all of the data won't fit on one diskette, use PCX-DOC2.BAT listed below, which uses XCOPY with the /M switch to copy only the files, that have changed, onto multiple diskettes:

```
@ECHO OFF
ECHO Back up data to a diskette
ECHO Please insert text diskette and press any key
PAUSE
XCOPY *.DOC A: /M
ECHO Please insert illustration diskette and press any key
PAUSE
XCOPY *.PCX A: /M
ECHO Data backed up
ECHO ON
```

You can find more information about backing up data in the corresponding chapter of this book. Remember that dividing the disk into specific directories and arranging data properly can make data security and working with application programs much easier.

Avoiding problems with certain MS-DOS commands

Certain MS-DOS commands are more prone to problems than others. Here are a few areas to watch out for:

Problems with APPEND

APPEND lets MS-DOS use a file even if it's not in the current directory without having to specify the entire path of that file.

Not all application programs nor even all MS-DOS commands (such as COPY, DEL or RENAME) can use the APPEND command. Some application programs are unable to process files that aren't in the current directory. For example, Microsoft Word 5.0 does not recognize the settings of the APPEND command.

Some application programs may also have problems with files in the search path. Although the program can read the file correctly, the file is saved in the current directory, not the original directory. If you notice an application program having these problems, don't use APPEND.

In general, APPEND shouldn't be used on directories with data that is processed. You can use APPEND for directories that contain the help files for a program (help texts, spell checker etc.).

Example

Your current directory is C:\PROG1 and you use APPEND to set a search path to the C:\DATA1 directory. This directory (C:\DATA1) contains the LETTER.DOC file. Start an application that uses the LETTER.DOC without specifying a path. Because APPEND was set, MS-DOS finds the file and loads it. However, when you save the file, the original version remains unchanged in the C:\DATA1 directory and a new modified version is saved in the C:\PRG1 directory. Not only have you wasted storage space, but since there are several versions of this file, you may accidentally work with an older version of the file rather than the newest version.

Problems with FASTOPEN

FASTOPEN speeds up access to the data on your hard drive by returning the physical disk location of files on the hard drive in memory. When a file is opened multiple times, its location is already in memory and therefore need not be retrieved from the hard drive again. This technique greatly speeds up retrieval of data from the file whenever applications use these files frequently.

Because of this technique, you may have problems with programs that change the location of a file on the hard drive (programs for hard drive optimization, such as SD.EXE from Norton Utilities, COMPRESS.EXE from PC Tools and BeckerTools Compress from Abacus). Don't use FASTOPEN while using these or similar programs. If different programs collide, it will not only crash your computer, but can also cause considerable loss of data.

Special features of TSR programs

Terminate and Stay Resident programs remain in the memory of the computer after being run. Usually you can activate this type of program by pressing a "hot" key.

Many driver programs (e.g., mouse drivers such as MOUSE.COM) are also kept in the memory after being started. You may encounter several problems when working with these kinds of programs:

Memory space

Each TSR or driver program reduces the amount of available memory in the computer. This can prevent large application programs from running or reduce the amount of available memory for processing data.

Sequence of programs

TSR programs change important information in the memory of your computer. These changes can cause problems, especially when you're using multiple TSR programs. Do not assign identical hot keys (key press combinations for activating a TSR program). Refer to the TSR's documentation pertaining to the sequence in which TSR's should be started.

Unfortunately, many software developers insist their TSR be started first. However, this can't always be done, specially if more than one TSR has this same requirement. You may have to experiment to find out which sequence of programs can work.

Hotkeys

Also remember that you can't use a key combination which calls a TSR program in any other application program. Doing this interferes with important program functions. For example, if a TSR program uses Alt + F to start, this key combination can't be used in Microsoft Word because it always activates the TSR program. Many TSR programs allow you to choose any key combination for starting when you install the program. Use this option to avoid conflicts with application programs.

Don't start TSR programs from within application programs

Many application programs let you use an MS-DOS command from within the application itself. When using Microsoft Word 5.0, for example, you can use the LIBRARY RUN command to execute an MS-DOS command. However, this command should never be used to install TSR programs.

For example, if you forgot to start the mouse driver for your word processor, quit the word processing program using the command for quitting, start the driver program and start the word processor again. Otherwise, you might have problems with memory management. You might not be able to start any other program after quitting the word processor.

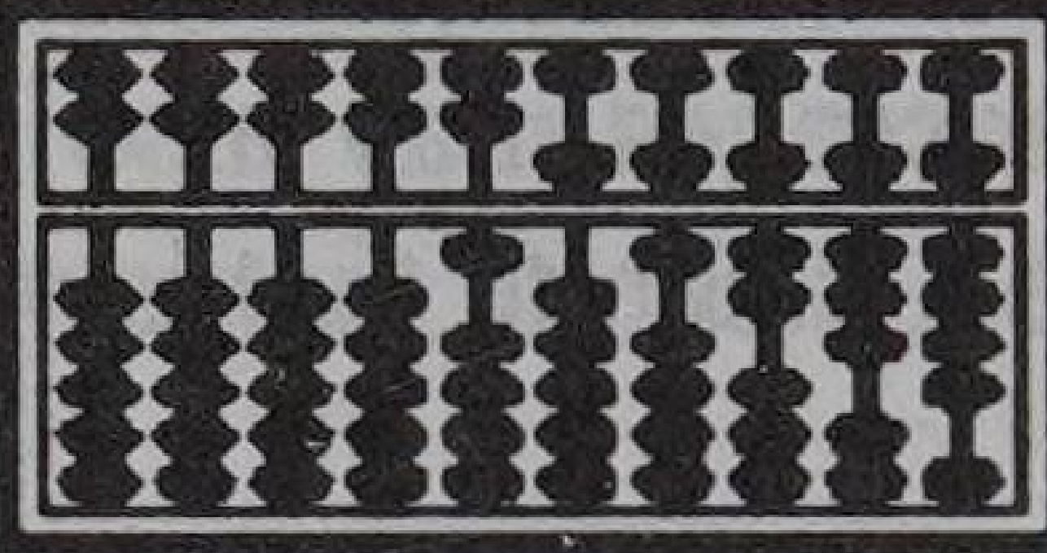
TSR programs in the MS-DOS Shell

You should remember some special features when working with the MS-DOS Shell. If you want to use a TSR program for your entire work session, start it before starting the MS-DOS Shell. For example, if you need to use the editing and macro capabilities of DOSKEY (supplied with MS-DOS 5.0), start DOSKEY in your AUTOEXEC.BAT file before starting the MS-DOS Shell.

Everything we've discussed about starting TSR programs from application programs also applies to the MS-DOS Shell. Using Shift + F9 or selecting MS-DOS Command Prompt leaves the MS-DOS Shell and goes to the command interpreter. Don't use this option to install TSR programs.

For example, if you forgot to start PRINT, press Alt + F4 to quit the MS-DOS Shell, start PRINT and then call the MS-DOS Shell a second time. Starting PRINT from within the MS-DOS Shell could cause problems with memory management. You might not be able to start any other programs after leaving the MS-DOS Shell.

Chapter



8

Data Input And Output

This chapter explains how MS-DOS communicates with users - how it inputs and outputs data. As a rule, MS-DOS commands accept your input from the keyboard and output messages to you on the screen.

However, you can change these defaults for any of a number of reasons. For example, instead of displaying a directory on the screen, you may want a hardcopy printout of it or you may want to sort data alphabetically before displaying them on the screen.

8.1 Setting Screen Output

The CLS command is used to clear the screen. Type CLS and press Enter and the data on the screen is cleared and the cursor is repositioned to the upper left corner of the screen.

MS-DOS also provides "filters", which are programs that process and change (filter) the screen output of MS-DOS commands. Filters enable you to sort data before it's displayed on the screen and to display information page by page, for example.

MORE (Output information page by page), SORT (Sort information) and FIND (Search for a text) are all MS-DOS filters. Unless you specify otherwise in the command line, these filters take the information to be processed from the current input device, which is usually the keyboard. The system prompt disappears when these filters are used.

You should know how these filters operate. As you probably already know, a command can be aborted by pressing Ctrl + C. You can also abort a filter command with Ctrl + C if you use one of the filters accidentally.

Since MS-DOS sets up a temporary file when you use a filter, there must be sufficient room for the file on the current storage device (normally, your hard drive) or the directory specified by the TEMP environment variable. If there is no TEMP variable and there isn't enough storage space to write the temporary files, MS-DOS displays the message:

```
File error during pipe
```

Other error messages may also be displayed for other reasons. For example, the drive might be a diskette that is write protected.

The filter command can still be used if another drive has the sufficient storage space. Make this drive your current drive and start the command from there. MS-DOS always creates the temporary file on the current drive.

Example

The current drive is A: and you want to use the following command to display the contents of the A: drive.

```
DIR | MORE
```

Since there isn't enough room on the diskette for the temporary file, MS-DOS displays an error message. However, there is enough room for the temporary file on a diskette in drive B:. So, type the following:

```
B:
DIR A:  | MORE
```

If you have a hard drive and want to avoid this problem permanently, MS-DOS 5.0 gives you the option of always creating these temporary files on a specific drive you select. Do this by defining an environment variable, named TEMP, which contains the complete path of the desired directory. For example, if you have a directory called C:\TMP and you want all temporary MS-DOS files to be placed in this directory, use the following command line:

```
SET TEMP=C:\TMP
```

If this line is added to your AUTOEXEC.BAT file, this definition is then activated whenever you switch on your computer.

Clearing the screen

To clear the screen and display the defined system prompt, use the CLS command as follows:

```
CLS
```

Using CLS can speed up screen output considerably. If the system prompt is located at the bottom of the screen and you want to display the directory of a disk, MS-DOS has to shift (scroll) the entire contents of the screen to the top for each new line that's displayed. Typing CLS before typing DIR clears the screen so that the display can begin at the top of the screen. This allows DOS to display the first 25 lines without wasting time scrolling the screen.

This technique can be used to speed up lengthy screen output, especially when working with batch files.

Displaying data one page at a time

The MORE filter can be used to display data one page at a time. MORE determines how many lines fit on the screen and then pauses at the end of each screen page. If the screen display has been altered by the MODE command, the MORE filter adjusts accordingly.

Displaying a file page by page

Use the following command to display a file page by page:

```
MORE <File
```

This command is quite useful when displaying lengthy files. Although you could use the TYPE command and press Ctrl + S to pause output to the screen, you might not be able to press Ctrl + S fast enough and thereby lose several lines that scrolled off the screen. When this happens, you must display the file again.

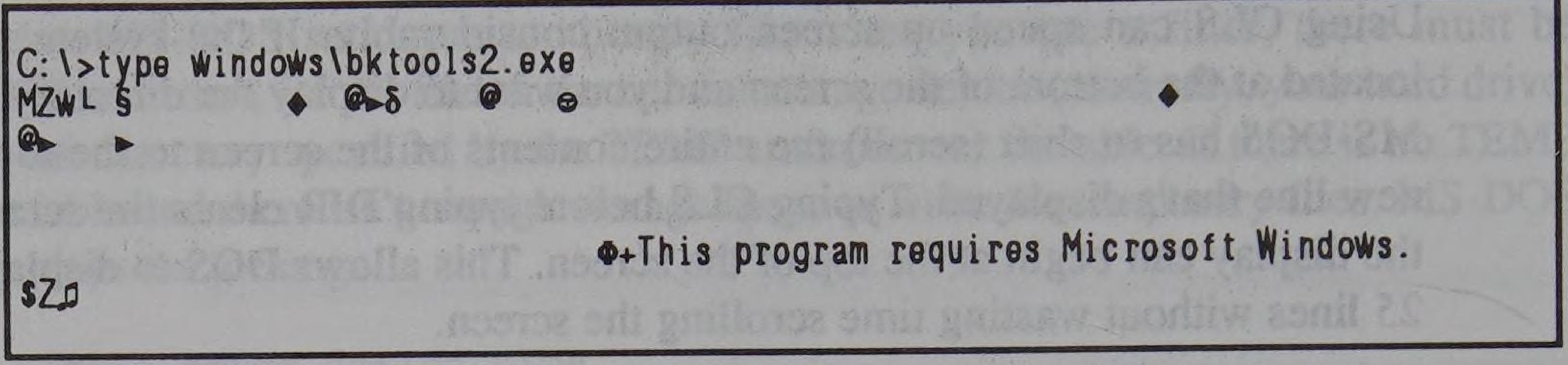
The redirection character "<", which redirects the input, must be used with the MORE command. This redirection indicates that instead of receiving its input from the keyboard, MORE receives input from the specified file.

After one screen page of the file is displayed, a message line appears at the bottom of the screen and MORE waits for a key to be pressed. After you press a key, it continues to display the next page.

You should display a file on the screen page by page only if it's an unformatted text file (an ASCII file). If you display programs or files from applications, strange characters will probably appear on your screen. Text files (ASCII files) and program files (binary files) are handled differently by DOS.

The end of an ASCII file is marked by the special end of file character (ASCII code 26, Ctrl + Z), while binary (program) files can end with any character. Binary files can contain end of file marks within a file.

So if you try to display a binary file on the screen by using TYPE or MORE, MS-DOS will stop when it encounters the first end of file mark. All you'll see on the screen are a few lines of strange characters.



```
C:\>type windows\bktools2.exe
MZwL §          ◆    @▸δ    @    ☺                         ◆
@▸      ▸


                                         Φ+This program requires Microsoft Windows.
$Z♫
```

Displaying a binary file with TYPE

If the file is very long and you want to stop displaying the file on the screen, press Ctrl + C. This key combination interrupts or aborts all MS-DOS commands.

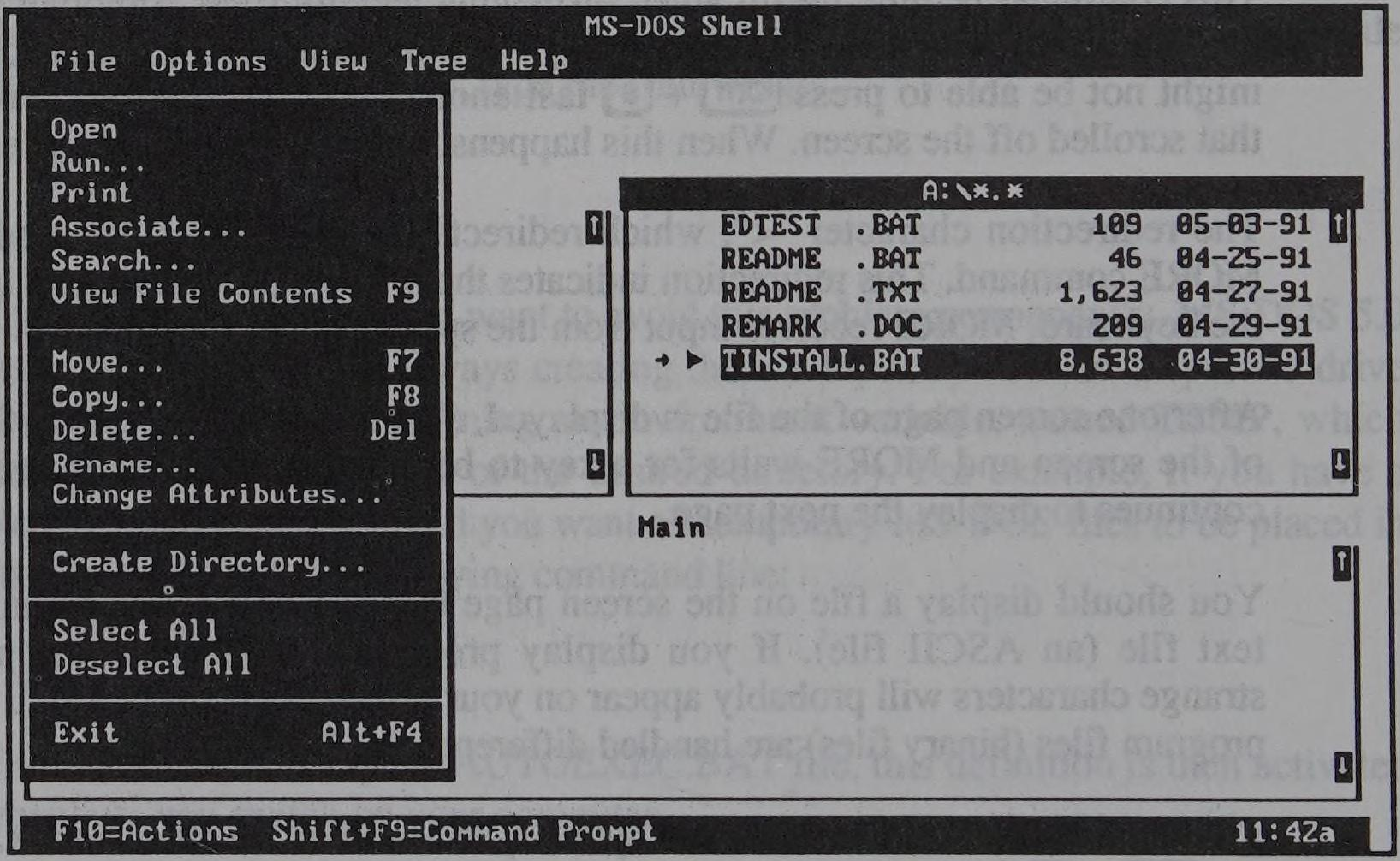
Example

To display the AUTOEXEC.BAT file page by page, use the command line:

```
MORE <C:\AUTOEXEC.BAT
```

If the file is shorter than a screen page, MS-DOS displays it on the screen without a message from MORE.

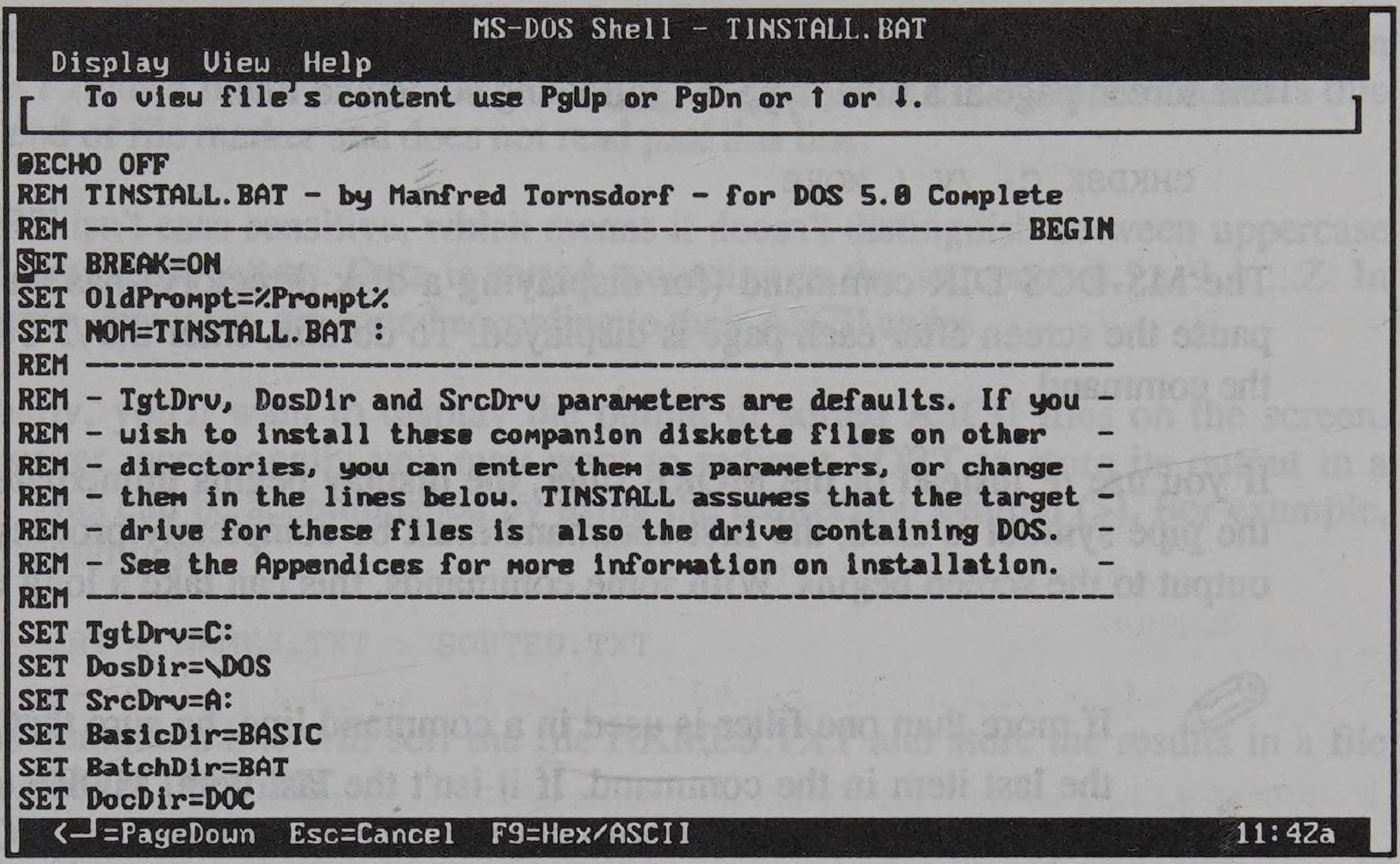
The MS-DOS Shell provides an easier way to display the contents of a file. Select the file from the file list then either select **View File Contents** or press F9.


MS-DOS Shell
File Options View Tree Help
Open
Run...
Print
Associate...
Search...
View File Contents F9
Move... F7
Copy... F8
Delete... Del
Rename...
Change Attributes...
Create Directory...
Select All
Deselect All
Exit Alt+F4
A:\*.*
Main
F10=Actions Shift+F9=Command Prompt
11:42a

```
EDTEST   .BAT      109  05-03-91
README   .BAT       46  04-25-91
README   .TXT    1,623  04-27-91
REMARK   .DOC      209  04-29-91
TINSTALL.BAT     8,638  04-30-91
```

Selecting the file TINSTALL.BAT from the file list and the View File Contents command from the File menu

When you select **View File Contents** (or press F9) a screen similar to the following appears which shows the contents of the selected file.


MS-DOS Shell - TINSTALL.BAT
Display View Help
<─┘=PageDown Esc=Cancel F9=Hex/ASCII
11:42a

To view file's content use PgUp or PgDn or ↑ or ↓.

```
@ECHO OFF
REM TINSTALL.BAT - by Manfred Tornsdorf - for DOS 5.0 Complete
REM ------------------------------------------------------- BEGIN
SET BREAK=ON
SET OldPrompt=%Prompt%
SET NOM=TINSTALL.BAT :
REM ------------------------------------------------------------
REM - TgtDrv, DosDir and SrcDrv parameters are defaults. If you -
REM - wish to install these companion diskette files on other   -
REM - directories, you can enter them as parameters, or change  -
REM - them in the lines below. TINSTALL assumes that the target -
REM - drive for these files is  also the drive containing DOS.  -
REM - See the Appendices for more information on installation.  -
REM ------------------------------------------------------------
SET TgtDrv=C:
SET DosDir=\DOS
SET SrcDrv=A:
SET BasicDir=BASIC
SET BatchDir=BAT
SET DocDir=DOC
```

Viewing the contents of the selected file from within the MS-DOS Shell

Displaying the screen output of other MS-DOS commands page by page

To display the screen output of an MS-DOS command one screen page at a time, enter the command with all of the necessary parameters. Then add the pipe symbol "|" and the MORE command:

```
Command parameters | MORE
```

MS-DOS first stores all the information which the command would have sent to the screen, in a temporary file and then redirects the input for MORE to this temporary file. When output is redirected through MORE, usually there is a brief pause before the first screen page is displayed.

If your keyboard doesn't contain the pipe symbol, you can type it by holding down the Alt key and pressing 1 2 4 on the numeric keypad. On extended keyboards, the pipe symbol is usually a separate key.

Example

The following command displays all of the files in the C:\DOS directory one screen page at a time:

```
DIR C:\DOS | MORE
```

The following command displays the directory tree on the hard drive C: one screen page at a time:

```
TREE C: | MORE
```

To display all of the files of the hard drive, along with their path specifications, one screen page at a time, type the following command line:

```
CHKDSK C: /V | MORE
```

The MS-DOS DIR command (for displaying a disk directory) has an option to pause the screen after each page is displayed. To do this, enter the /P switch after the command.

If you use /P instead of the MORE filter, the display begins immediately. When the pipe symbol is used, the first command must be completely processed before output to the screen begins. With some commands, this can take a long time.

If more than one filter is used in a command line, be sure that MORE is the last item in the command. If it isn't the last item, MORE will pause its processing after each screen page and the commands after MORE won't receive the information until you press a key.

Displaying the contents of files with TYPEM.BAT

By using the following batch file, you won't have to enter the pipe symbol. This batch file is contained on the companion diskette in the BAT directory. This batch file, called TYPEM.BAT, can be used in place of the TYPE command:

```
TYPE %1 | MORE
```

The following file, called TYPEM2.BAT, is a more extensive version of TYPEM.BAT:

```
@ECHO OFF
IF NOT "%1"=="" GOTO output
ECHO You must specify a filename
GOTO End
:output
TYPE %1 | MORE
:End
```

Sorted output

The filter SORT is used to display files and screen output sorted alphabetically. The size of the data files is limited to 63K. Larger files will generate an error message.

Displaying sorted files

Use the following command line to display a file in alphabetical sequence:

```
SORT <File
```

SORT looks at all of the characters in a line to determine the sort sequence. When SORT finds a line with the character Ctrl + Z (ASCII code 26), it considers this the end of file marker and does not read past this line.

SORT isn't case sensitive, which means it doesn't distinguish between uppercase and lowercase letters. Data is sorted according to the sequence 0,1,...9,A,...Z. In general, characters are sorted according to their ASCII codes.

Usually, you'll want to display the output of sorted ASCII files on the screen. However, occasionally you may want to redirect SORT to store its output in a file. This can be accomplished by using the redirection symbol (>). For example, you could type:

```
SORT < NAMES.TXT > SORTED.TXT
```

This command line will sort the file NAMES.TXT and store the results in a file named SORTED.TXT.

Example

Although you're unlikely to want to sort and display your AUTOEXEC batch file, we're using in this example because most users include an AUTOEXEC batch file on their system. Type the following command:

```
SORT <C:\AUTOEXEC.BAT
```

Large sorted files can be displayed a page at a time by adding the MORE filter:

```
SORT <C:\AUTOEXEC.BAT | MORE
```

Sorting and displaying the output of other MS-DOS commands

To sort and display the results of other MS-DOS commands, type the command with all necessary parameters and add the pipe symbol "|" followed by the SORT command:

```
Command parameters | SORT
```

MS-DOS writes all screen output from the first command into a temporary file. Then the file is redirected to the input of SORT. There may be a slight delay while the command line is processed because the temporary file must be created and the first command must be completely executed.

Example:

Typing TYPE to display the AUTOEXEC.BAT file and then sorting the display with SORT will produce the same results as the preceding example. The SORT filter processes the screen output of the MS-DOS TYPE command in the following example.

```
TYPE C:\AUTOEXEC.BAT | SORT
```

Special sorting switches

/R	Specifying the /R switch reverses the sorting direction (sort order). So when this switch is specified, the sort order is from Z to A and from 9 to 0.
/+n	When /+n is specified, SORT begins sorting according to characters in the nth column. If this switch isn't specified, SORT begins sorting at the first character in each line.

Example: Displaying a disk directory sorted alphabetically

You can use the following command to sort and display all of the files in the DOS directory:

```
DIR C:\DOS | SORT
```

It's easier to read the data if you display it one screen at a time. To do this use the following command:

```
DIR C:\DOS | SORT | MORE
```

The filenames and every line displayed by the DIR command, including the information about the storage device, is sorted alphabetically.

The DIR command has its own switch for sorting data. The files in the DOS directory can be sorted by filename and displayed by using the following command:

```
DIR C:\DOS /O:N
```

To display the data page by page, simply add the /P switch for page:

```
DIR C:\DOS /O:N /P
```

Example: Disk directory sorted by the filename extension

To sort all of the files in the DOS directory according to file extension and display them one screen page at a time, use the following command:

```
DIR C:\DOS | SORT /+10 | MORE
```

MS-DOS 5.0 also lets you accomplish the same result as the above command by using a switch with the DIR command.

```
DIR C:\DOS /O:E /P
```

In addition to files, DOS 5.0 also displays subdirectories and additional information about the storage device. The DIR command also sorts these subdirectories and interrupts the display when you use the SORT filter to sort. Use the sorting options of the MS-DOS DIR command for sorting by filename, extension, etc. Only DIR contains these options.

Example: Displaying files sorted according to attributes

In the following example, SORT is very useful because the MS-DOS command ATTRIB doesn't use any special "sorting parameters". The following command displays all of the files in the current directory sorted according to file attributes:

```
ATTRIB *.* | SORT
```

Displaying only lines that contain specific text strings

The FIND command can be used to limit the display to either the lines from the file which contain certain text strings or the lines that don't contain the strings.

Although FIND was case sensitive in previous versions of DOS, MS-DOS 5.0 includes a new switch (/I), which ignores the case of characters.

The following command displays only the lines that contain the search string:

```
Command parameter | FIND "search string"
```

The search string must be enclosed in quotation marks. If the character sequence for which you're searching is already enclosed in quotation marks, type another set of quotation marks ("").

Example

To display all of the files, in the C:\DOS directory, that were created or changed in 1989, use the following command:

```
DIR C:\DOS | FIND "-89"
```

The minus sign (or dash), which is placed in front of 89, is important because it limits the displayed files to those that have the date sequence "xx-xx-89".

The FIND filters help sort and display disk directories without having the data placed on the storage device. You may want to use the following short batch file, DIRS.BAT, to sort and display directories:

```
@ECHO OFF
DIR %1 | FIND /v "e" | SORT | MORE
```

You can specify a file after DIRS, DIRS *.EXE or DIRS T*.TXT. You could also specify any directory, such as DIRS C:\DOS. If you don't specify any parameters, the command sorts and displays all of the files in the current directory.

After DIR is called, the "e" eliminates (filters) the information that is usually displayed at the top and bottom of the screen, such as the number of files, bytes used and bytes free information. Only the names of the files and corresponding information will be displayed.

The special /V switch reverses the search condition and only lines that don't contain the search string are displayed. Since all of the files and directories are displayed in capital letters and the disk information lines each contain a lowercase "e", the batch file prevents lines that have an "e" from being displayed.

It's also possible to use the filters one after another. The single command elements are separated by the pipe symbol. The example above also illustrates the meaning of the term "pipe". The data generated by the DIR command are processed and handled as if they were passing through a pipe before the information is displayed on the screen (after MORE).

If more than one filter is used in a command line, be sure that MORE is the last item in the command. If it isn't the last item, MORE will stop the "processing line" after every screen page and the commands following MORE won't receive the information until you press a key.

Using FIND to search for files

To search the entire hard drive for a specific file, use the following command:

```
CHKDSK drive /V | FIND "FILENAME"
```

In uppercase letters, specify the filename or the known part of the name. This is necessary because DIR outputs filenames in uppercase letters and the FIND command differentiates between upper and lowercase letters. As explained below, an optional switch, /I, can be used with FIND to ignore the case of characters during the search. The command would be:

```
CHKDSK drive /V | FIND /I "FILENAME"
```

Special search switches

By specifying additional switches with the FIND command, you can use the special switches that are available in the search. Some of these switches are ignored if FIND is used after a command as a filter:

/V	Reverses the search condition; displays only those lines that do not contain the search string.
/C	Displays only the number of lines that contain the search string.
/N	Displays the line number of each line that contains the search string.
/I	This new switch ignores the case of characters in the search string.

Example: Counting the lines of the AUTOEXEC.BAT

To display all the lines in the AUTOEXEC.BAT file with line numbers, use the /N and /V switches and specify a search string that doesn't appear in any line of AUTOEXEC.BAT:

```
FIND /N /V "abcdef" C:\AUTOEXEC.BAT
```

With this command line, FIND counts all of the lines that don't contain the search string. In this example, this is all of the lines in the AUTOEXEC.BAT file.

Example: A better way to sort a disk directory

When the SORT filter is used to sort a disk directory, it also sorts the names of subdirectories and the additional data about the storage device. This information may not be necessary when redirecting the directory to a file. If you don't want this information to appear, use FIND.

The following command line displays and sorts only the files in the current directory by name. (Ensure that there are twelve spaces before the zero.):

```
DIR | FIND /V "<DIR>" | FIND /V "e" | FIND /V "            0"
| SORT | MORE
```

To sort the files by extension, use SORT /+10, to sort by file size, use SORT /+13. Let's look at this line in detail:

Explanation

As you can see, this command line is rather complex. When you're working with command lines like this you should create simple batch files that contain the command line or use the DOSKEY command of MS-DOS 5.0.

By using these methods, you won't have to enter the complete line every time you want to use the command or after making a typing error while entering the line.

Each element of the command has the following meanings:

```
FIND /v "<DIR>"
```

This statement limits the search to only files by excluding lines that contain the search string "<DIR>".

```
FIND /v "e"
```

Next, FIND is used again to exclude information about the storage device (bytes free, bytes used, etc.). As previously explained, DIR outputs filenames in uppercase letters, while the lines of the additional information each contain at least one lowercase "e" (make sure there are twelve spaces before the zero).

```
FIND /v "            0"
```

This part of the command looks rather strange. When MS-DOS passes the information from DIR to MORE using FIND and SORT, temporary files are created and then deleted.

These temporary files don't have extensions and are 0 bytes long when the DIR command is used, they are deleted by the DIR command.

```
SORT
```

The SORT command sorts the data. By specifying /+xxx, the command will sort the files by extension (where xxx = 10) or by file size (where xxx = 13).

```
MORE
```

MORE displays only one screen page of data at a time. This is very useful with long directories. As explained earlier, MS-DOS waits for you to press a key before displaying the next screen of information.

When using DIR, you cannot use both MORE and the /P switch to display the output page by page.

Depending on the size of the directory, you may need to press a key a few times to continue the DIR command (if /P is used). When the line is processed, all of the information will quickly scroll by on the screen.

Each time the output from the DIR command completely filled the screen, the message "Press any key to continue..." would be displayed. MS-DOS doesn't pass the information from the first command DIR to FIND until DIR has displayed all of the information.

Display directories sorted alphabetically

The following command line outputs only the subdirectories in alphabetical order. Make sure there are twelve spaces before the zero. It also displays "." and ".." because these symbols represent the current directory and the parent directory:

```
DIR | FIND "<DIR>" | FIND /V "e" | FIND /V "            0" |
SORT | MORE
```

The MS-DOS 5.0 DIR command not only sorts, but can also limit the display to certain kinds of files or directories. There is an easier way to obtain the same results (without the MS-DOS SORT filter):

```
DIR /a:d /o:n /p
```

8.2 Redirecting Input And Output

Usually MS-DOS commands receive their instructions and information from the keyboard and output information to the screen. You can change these defaults when you enter the command for the DOS command.

This is called redirecting the input or output. Redirection can help reduce a lot of tedium and typing in your daily work.

Redirecting input to a file

MS-DOS commands usually receive input from information typed at the keyboard. To change this, use the "<" character to redirect the input.

To direct the input from a file to a command, use the following command:

```
COMMAND <FILENAME
```

Parameters can be specified after the command. The command will then take its input from the specified file and process it as if the input was being typed in from the keyboard.

A command that has had its input redirected to a file receives no information from the keyboard. Complete control is given to the specified file and doesn't return to the keyboard until the command has been completed. The command can't be aborted if the file doesn't contain any characters for ending the command.

If you encounter problems, you must perform a warm boot (press the Ctrl Alt Del keys). Unfortunately, you could lose data when this happens. The special MS-DOS filters are an exception to this rule (MORE, SORT and FIND) because they process a file to the end and then quit.

Input can only be redirected for commands that receive input from the default input device. You cannot redirect the input of commands that read the keyboard directly.

Example: Displaying the date

Every time your computer is started, the current date should be displayed. However, you don't want this to interrupt the execution of your AUTOEXEC.BAT file. So the DATE command must obtain the input it needs (ENTER) from a file so that it can end without changing the date.

Use COPY CON to create a file called ENTER.DAT in the C:\DOS directory. The file should contain only a blank line:

```
COPY CON C:\DOS\ENTER.DAT
```

Press Enter at the beginning of the following line. Press Ctrl + Z at the end of the line and press Enter again. Test the following line and then include it in your AUTOEXEC.BAT file:

```
DATE <C:\DOS\ENTER.DAT
```

The date is displayed and the system prompt immediately appears. The DATE command took input from the ENTER.DAT file.

Example: Displaying the boot sector of a diskette

Input redirection can also be used to control complex processes automatically. The EDLIN editor and the DEBUG program, which is used for finding errors, are ideal candidates for input redirection because they can process more than one line of input.

The following example can be`used to display the first sector (boot sector) of a diskette in drive A:. Create an ASCII file using EDLIN or another editor. This file, called BOOT.REC, should contain the following lines:

```
L 0100 0 0 1
D 0100 01FF
Q
```

This file contains the exact commands to instruct DEBUG to load the boot sector into memory, display it on the screen and automatically stop. In order for DEBUG to execute these commands, redirect the input from BOOT.REC to DEBUG:

```
DEBUG <BOOT.REC
```

You could also create a batch file, called SHOWBOOT, which contains the preceding command line.

Redirecting output to a new file

Normally MS-DOS commands display information on the screen. The characters ">" and ">>" can be used to redirect output to a specific device or file.

To write all of the output from a command to a file instead of displaying it on the screen, use the command:

```
COMMAND PARAMETERS >FILENAME
```

Parameters can be specified with the command. MS-DOS creates a file with the specified filename and redirects all output from the command to this file. An existing file with the same name as the specified filename is overwritten.

The output of some commands cannot be redirected because the commands don't output information to the default output device, (the screen). Instead they send their output to a special device for error handling.

For example, some programs write information directly to video memory. You cannot use the redirection character to redirect the screen output of these programs to a file.

Example

If you use Turbo Pascal to develop programs and want to redirect the screen output to a file, don't use the CRT unit in programs. By doing this, the screen output of the program goes to the standard output device and can be redirected. Not using the CRT unit does slow down the speed of output. If you absolutely must use the CRT unit, use the special AssignCrt command in Turbo Pascal.

Example

You want to record the time whenever your computer is switched on. For this example, you'll need the ENTER.DAT file, which was created in the previous section. Add the following line to your AUTOEXEC.BAT:

```
TIME <C:\DOS\ENTER.DAT >TIME.DAT
```

This causes the TIME command to store the current time in the TIME.DAT file every time you boot your computer. This file can be viewed, after restarting your computer, by typing:

```
TYPE TIME.DAT
```

Adding output to an existing file

Occasionally you may want to combine the output of several commands into a single file. The redirection sequence ">>" can be used instead of the ">" character:

```
COMMAND >>FILENAME
```

MS-DOS adds the output of this command to an existing file. If the file doesn't exist yet, MS-DOS creates it.

This method can be used to record how many hours you work on your computer. In a file called TIME.DAT, simply enter the times you start and stop working each day. Add the following command lines to your AUTOEXEC.BAT file or in a batch file named STARTWRK.BAT to record your starting times:

```
ECHO ***** Start **** >>C:\DOS\TIME.DAT
DATE <C:\DOS\ENTER.DAT >>C:\DOS\TIME.DAT
TIME <C:\DOS\ENTER.DAT >>C:\DOS\TIME.DAT
```

These batch files are contained on the companion diskette in the BAT directory. At the end of your work session, call another batch file named ENDWRK.BAT before switching off your computer to record your quitting time. This batch file writes the time you stop working into the same file:

```
ECHO ***** End of work **** >>C:\DOS\TIME.DAT
DATE <C:\DOS\ENTER.DAT >>C:\DOS\TIME.DAT
TIME <C:\DOS\ENTER.DAT >>C:\DOS\TIME.DAT
```

Example: Simple directory change

By redirecting additional screen output to a file, you can create new command lines from existing commands and screen output. For example, a batch file that automatically stores the current directory and permits recalling it instantly has the following requirements:

1. A file must be created in the DOS directory that contains only the CD command and a space. (Press the Shift key and the - key (minus) to create the underscore character in the following command line.)

   ```
   COPY CON C:\DOS\_CD.BAT
   ```

   Press the Enter key, then press the keys C D Spacebar, Ctrl + Z and finally Enter.

2. To extend this batch file so it contains the desired command line, we add the screen output of the CD command to the batch file. For example, if C:\DATA\DOC is the current directory, the command would be:

   ```
   CD >>C:\DOS\_CD.BAT
   ```

   Type:

   ```
   CD C:\DOS
   ```

This command line will switch your current directory to the DOS directory. Use TYPE and the complete pathname to display the _CD.BAT file. It now contains the command, CD and the name of the previous directory (in this example, C:\DATA\DOC). To make C:\DATA\DOC the current directory again, type:

```
_CD
```

By using these lines from batch files, you can use one command to store the current directory and then another command to return to this directory at any time. To make this even easier, recreate the _CD.BAT in your DOS directory and name it to _CD.DAT. This file should only contain the CD command and a space.

Then create another file called NOTE.BAT in your DOS directory. This batch file is contained on the companion diskette in the BAT directory. This file must have the following contents:

```
COPY C:\DOS\_CD.DAT C:\DOS\_CD.BAT
CD >> C:\DOS\_CD.BAT
```

Now, when you want to store the name of the current directory, simply type NOTE. This causes the batch file NOTE.BAT to create another batch file, _CD.BAT, which stores the information for returning to the current directory. Now you can change to any directory and return to the previous directory by typing _CD.

This command can be further expanded by using a variable for the pathname of the directory that should be made current. Type the following lines to create a new batch file, called ACD.BAT, which will automate the entire storing and changing process:

```
ECHO OFF
COPY C:\DOS\_CD.DAT C:\DOS\_CD.BAT
CD>>C:\DOS\_CD.BAT
CD %1
ECHO ON
```

Remember to type Ctrl + Z to save the new batch file. This batch file (ACD.BAT) will automatically store the name of the current directory, change to a new directory and enable you to return to the previous current directory by typing _CD.

Remember that the entire pathname must be specified to change directories. For example, type the following to change from the BAT directory to the DOS directory.

```
ACD C:\DOS
```

Redirecting output to the printer

It's easy to redirect screen output to the printer:

```
Command >LPT1
```

If your printer isn't connected to the first parallel port (LPT1), use the name for the appropriate printer connection.

Example: Printing a disk directory

The following can be used to print the current disk directory:

```
DIR >LPT1
```

Example: Documenting the execution of a command

To copy all of the files in the current directory to a diskette in drive A: and document the backup process by printing it, use the following command line:

```
COPY *.* A: >LPT1
```

Example: Printing a very compact disk directory

To find the contents of a diskette any time, print the disk directory. However, if you want to attach long directories to the diskette sleeve, the standard output can be very long.

The following batch file, PRDIR.BAT, creates the smallest possible printout of the contents of any drive:

```
@ECHO OFF
REM Print compact disk directory to dot matrix printer
MODE LPT1:132,8 > NUL
DIR %1 /W | FIND /V "e" >LPT1
MODE LPT1:80,6 > NUL
ECHO ON
```

This batch file is contained on the companion diskette in the BAT directory. You must pass the drive as a parameter when calling the batch file, as follows:

```
PRDIR A:
```

The batch creates a 5-column disk directory, removing all extra information about the diskette (volume and volume number, free space on the diskette). In addition, the batch switches to the smallest, most narrow font possible. Since the printout is still small enough for floppy diskettes, you can easily attach it to the diskette sleeve.

This batch file only works with dot matrix printers.

8.3 Linking Commands

Several commands can be linked together so that one command passes its output to the next command, which uses it as input. Use the pipe symbol "|" to separate the commands to be linked.

```
Command1 | Command2 | Command3
```

Command1 receives its input from the keyboard. Then DOS passes the screen output of command1 to command2. Command3 receives the screen output of command2 as its input and the output from command3 is displayed on the screen.

Linking several commands is especially useful when you're using filters.

Input and output can also be redirected. For example, you can take the input for the first command from a file instead of the keyboard or redirect the output of the last command to the printer instead of the screen.

Example

With the following command, the current directory is sorted and displayed one page at a time:

```
DIR | SORT | MORE
```

The next command outputs the current diskette directory to the printer, sorted and one page at a time:

```
DIR | SORT | MORE >LPT1
```

MS-DOS filters are not only used when linking commands. You could also use the output of the ECHO command as input for DEL.

➡ The next two examples illustrate this. Since the command lines delete data, use them only after carefully considering the results. Be careful you don't accidentally delete useful files.

The following command line deletes the contents of the current directory (except files specified with the ATTRIB command) without prompting you for confirmation. You can use it to delete data within a batch file automatically:

```
ECHO Y | DEL *.*
```

The DEL command receives the necessary confirmation of Y + Enter from the ECHO command.

The following batch file, DELDIR.BAT, removes all files from any specified subdirectory and then removes the subdirectory. This batch file is contained on the companion diskette in the BAT directory.

```
ECHO Y | DEL %1
RD %1
```

For example, if the batch file is in the C:\BAT directory (specified by the search path for commands) and the C:\SCRAP directory contains files that are no longer needed, the following command line can be used to remove the files and the directory:

```
DELDIR C:\SCRAP
```

The character sequence "%1" allows you to pass parameters that can be used in the batch file.

Chapter

9

System Settings

MS-DOS is designed to be flexible. As the user adds new hardware, DOS can be changed to accommodate the new devices. Just as importantly, DOS can be "customized" or tailored for the work habits or preferences of the individual user.

For example, the system prompt can be changed, the number of buffers altered to speed up access to the hard drive and special function programs, such as the MS-DOS-Shell or DOSKEY, can be installed.

DOS can be adapted to the computer and your special needs by modifying the two system files AUTOEXEC.BAT and CONFIG.SYS.

The best way to change these files is to use the DOS 5.0 editor, EDIT. Make back up copies of the original files before modifying them so that you can use these copies in the event that one of the changes you make creates problems.

You should also create a bootable diskette (use the FORMAT /S command). Then if one of the changes you make to the CONFIG.SYS file prevents your computer from booting properly from the hard drive, you can use the bootable floppy diskette. You'll simply insert the bootable diskette into a disk drive and restart the computer; then recopy the original CONFIG.SYS file to the hard drive.

9.1 Setting The Time And Date

As part of its task of maintaining files and directories, DOS keeps track of the date and time at which a file or directory is created or changed.

It's important that the date and time be accurate. If your computer doesn't have a built-in clock, the correct time and date must be entered immediately after you start the computer. Since these settings must be made each time you start your computer, you should include the DATE and TIME commands in your

AUTOEXEC.BAT file, which automatically runs every time the computer is started. These two commands will prompt you to enter the date and time.

Since most computers nowadays do have built-in clocks, you may not have to use the DATE and TIME commands to manually set it. In this case, the built-in clock automatically maintains the correct date and time. The DATE and TIME commands are then used to alter the date and time for certain situations (e.g., set to daylight savings time).

Some computer clocks lose or gain several minutes per day. So even if your computer has a battery operated clock, check the date and the time regularly.

Displaying and changing the date

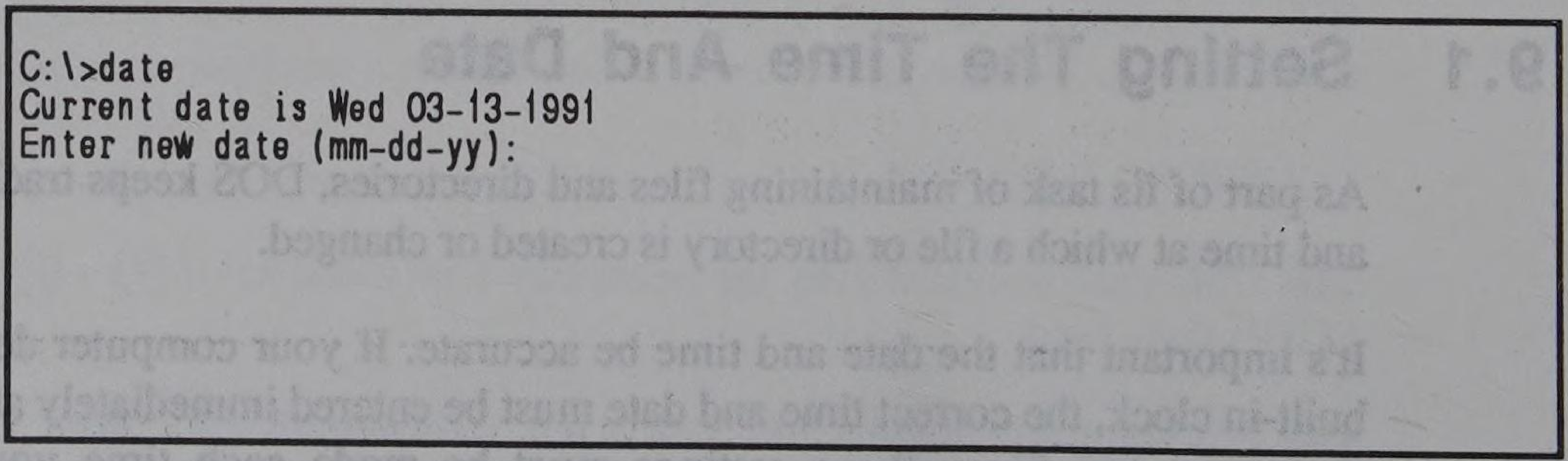
Use the DATE command to display and make any necessary changes to the current date:

```
DATE
```

After pressing Enter, the current date that is set in your computer is displayed along with a prompt that asks you to enter a new date. Enter the new date in the MM-DD-YY format (or DD-MM-YY if you've specified a country other than the U.S. in your AUTOEXEC.BAT file):

MM	Month (1-12)
DD	Day (1-31)
YY	Year (80-99) or (1980-2099)

Instead of a hyphen "-", you can also use a period "." or a slash "/" to separate the elements of the date. The year can be specified as either two or four digits (91 or 1991).



```
C:\>date
Current date is Wed 03-13-1991
Enter new date (mm-dd-yy):
```

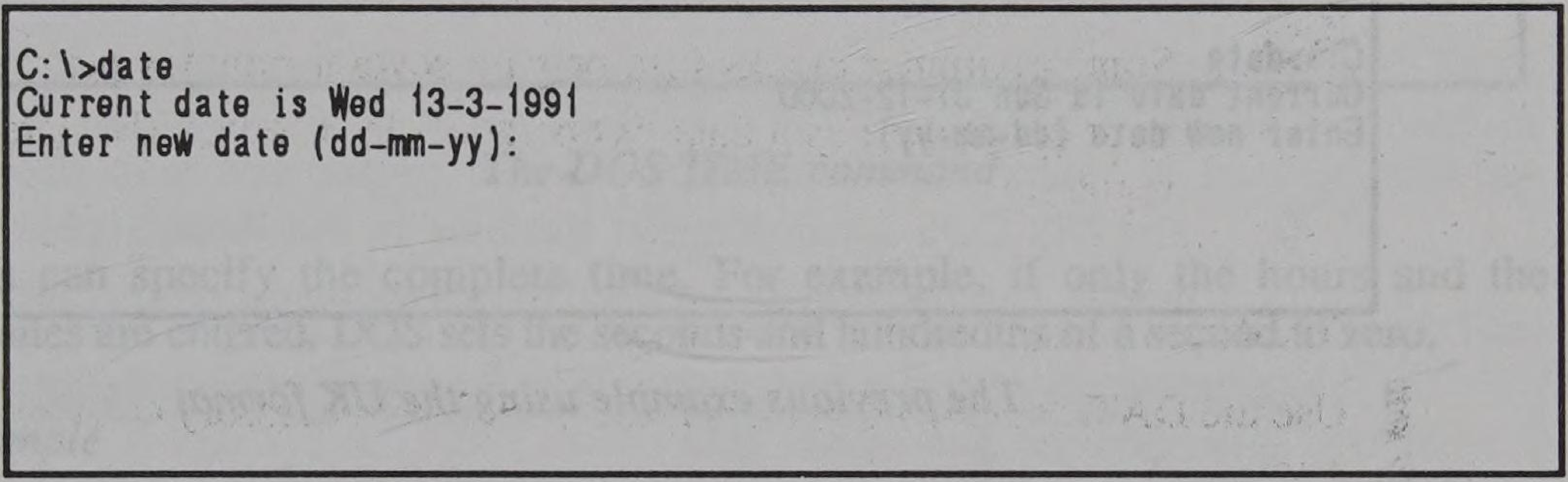
The DOS DATE command

The new date can also be passed as a parameter to the DATE command. Use the same format as described above when entering the date. For example, to set 6-20-1991 as the current date, use the following:

```
DATE 6-20-91
```

Depending on the date format (e.g., UK), you may need to type:

```
DATE 20-6-91
```



```
C:\>date
Current date is Wed 13-3-1991
Enter new date (dd-mm-yy):
```

The DOS DATE command using UK format

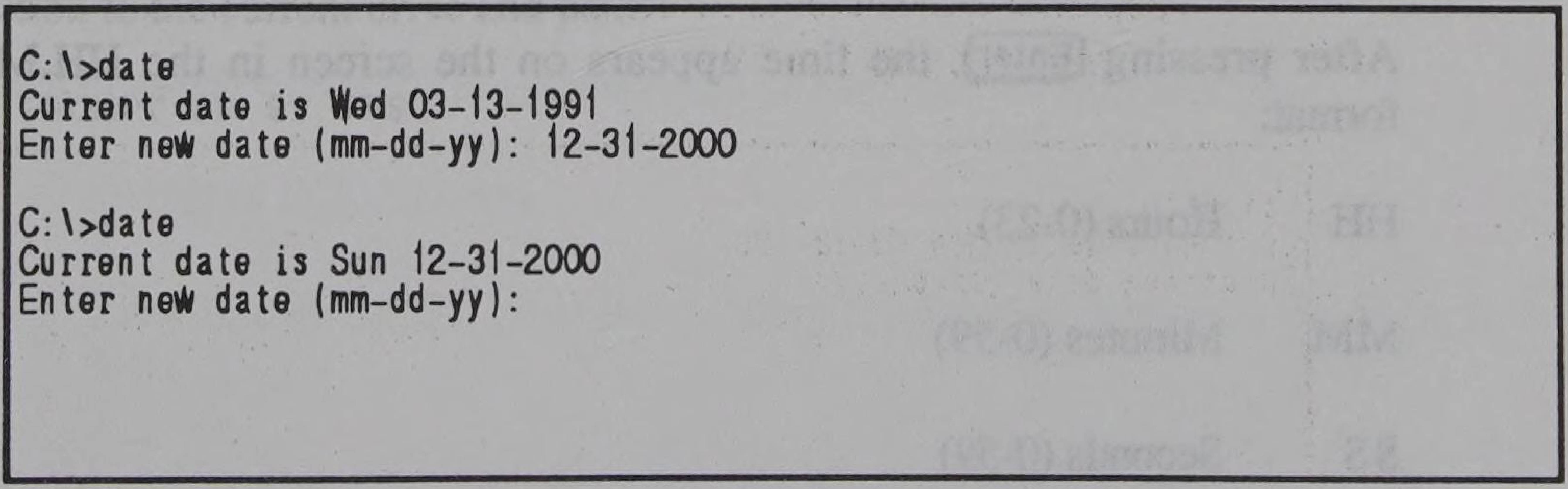
In addition to the date, DOS displays the day of the week. So you can use this function to determine on which day of the week a certain date falls. To do this, simply enter the date as a parameter to the DATE command.

Remember to reset the current date before continuing your work on the computer. If you don't do this all of the new or changed files will have an incorrect date stamp in the directory.

Example

To determine on which day of the week New Year's Eve will fall in the year 2000, type the following date after calling DATE:

```
12-31-2000
```



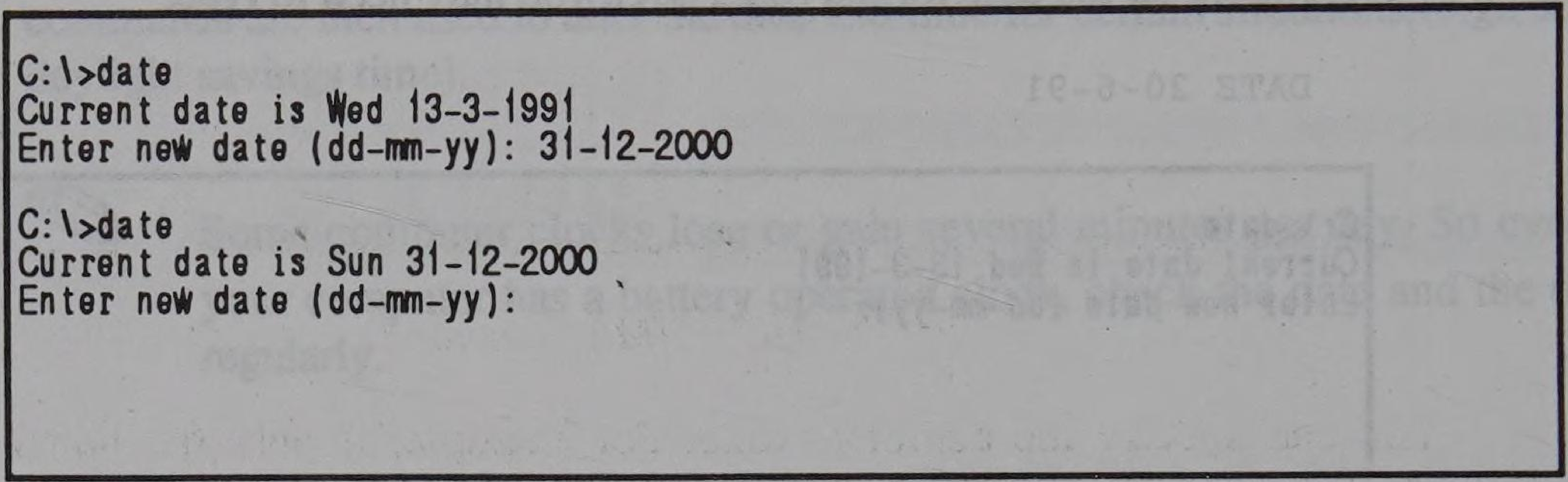
```
C:\>date
Current date is Wed 03-13-1991
Enter new date (mm-dd-yy): 12-31-2000

C:\>date
Current date is Sun 12-31-2000
Enter new date (mm-dd-yy):
```

The DATE command displays that December 31, 2000 is a Sunday

After you type the DATE command again, you'll see that the last day of the year falls on a Sunday.

Depending on the format (e.g., UK), you may need to type:

```
31-12-2000
```



```
C:\>date
Current date is Wed 13-3-1991
Enter new date (dd-mm-yy): 31-12-2000

C:\>date
Current date is Sun 31-12-2000
Enter new date (dd-mm-yy):
```

The previous example using the UK format

Now you can reset the date to the current date. Otherwise, if you press Enter instead of entering another date, the previous date will be set.

If you try to enter an invalid date, DOS displays an error message informing you of this and prompts you to enter a new date.

Viewing and changing the time

Use the TIME command to display and change the time stored by your computer. If your computer doesn't have a battery operated clock, include the TIME command as well as the DATE command in your AUTOEXEC.BAT file.

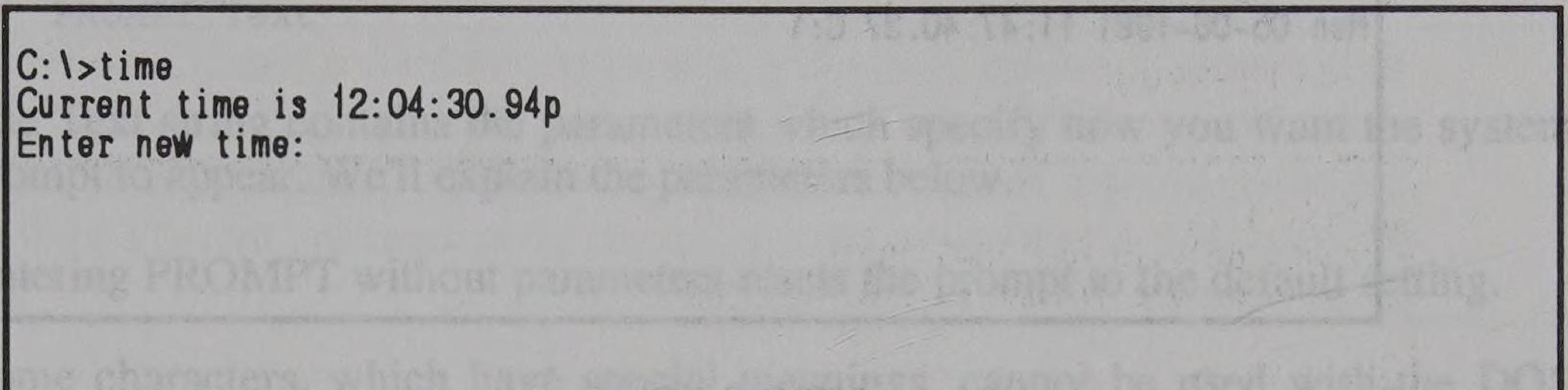
If a built-in clock isn't installed, DOS sets the clock to 00:00:00.00 (0 hours, 0 minutes, 0 seconds, 0 hundredths of a second) each time you switch on the computer.

```
TIME
```

After pressing Enter, the time appears on the screen in the HH.MM.SS.XX format:

HH	Hours (0-23)
MM	Minutes (0-59)
SS	Seconds (0-59)
XX	Hundredths of a second (0-99)

If a new time isn't needed, simply press Enter; the time remains unchanged. If you enter an invalid time, DOS indicates this with an error message and prompts you to enter a new time.



```
C:\>time
Current time is 12:04:30.94p
Enter new time:
```

The DOS TIME command

You can specify the complete time. For example, if only the hours and the minutes are entered, DOS sets the seconds and hundredths of a second to zero.

Example

Type 17:10; DOS interprets your entry as 5:10:00.00 PM.

DOS accepts the time entered in either the military format (24 hour) or in the standard 12 hour notation with an "A" or "P" (AM and PM) added to the time string. So the time in the preceding example could also be set by typing:

```
TIME 5:10P
```

Displaying the date and time on the command line

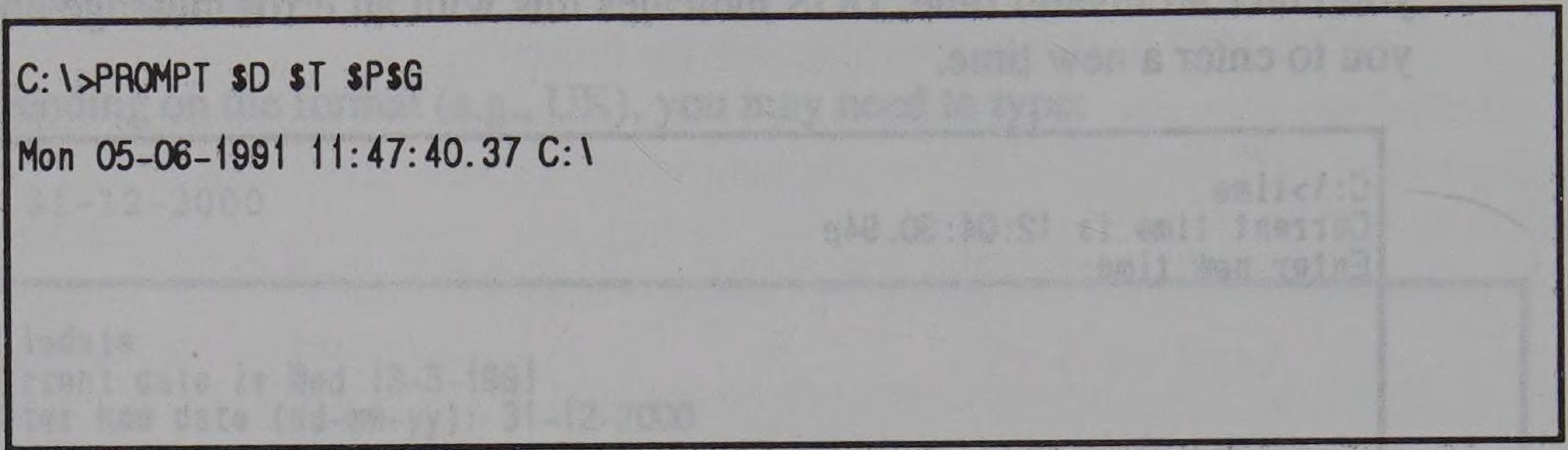
The date and time can be displayed as part of the system prompt. Use the DOS PROMPT command to include the following characters:

```
$d $t
```

The character sequence $d specifies the date, while $t specifies the time. The following command sets the system prompt to display both the date and time in addition to the current drive and path:

```
PROMPT $D $T $P$G
```

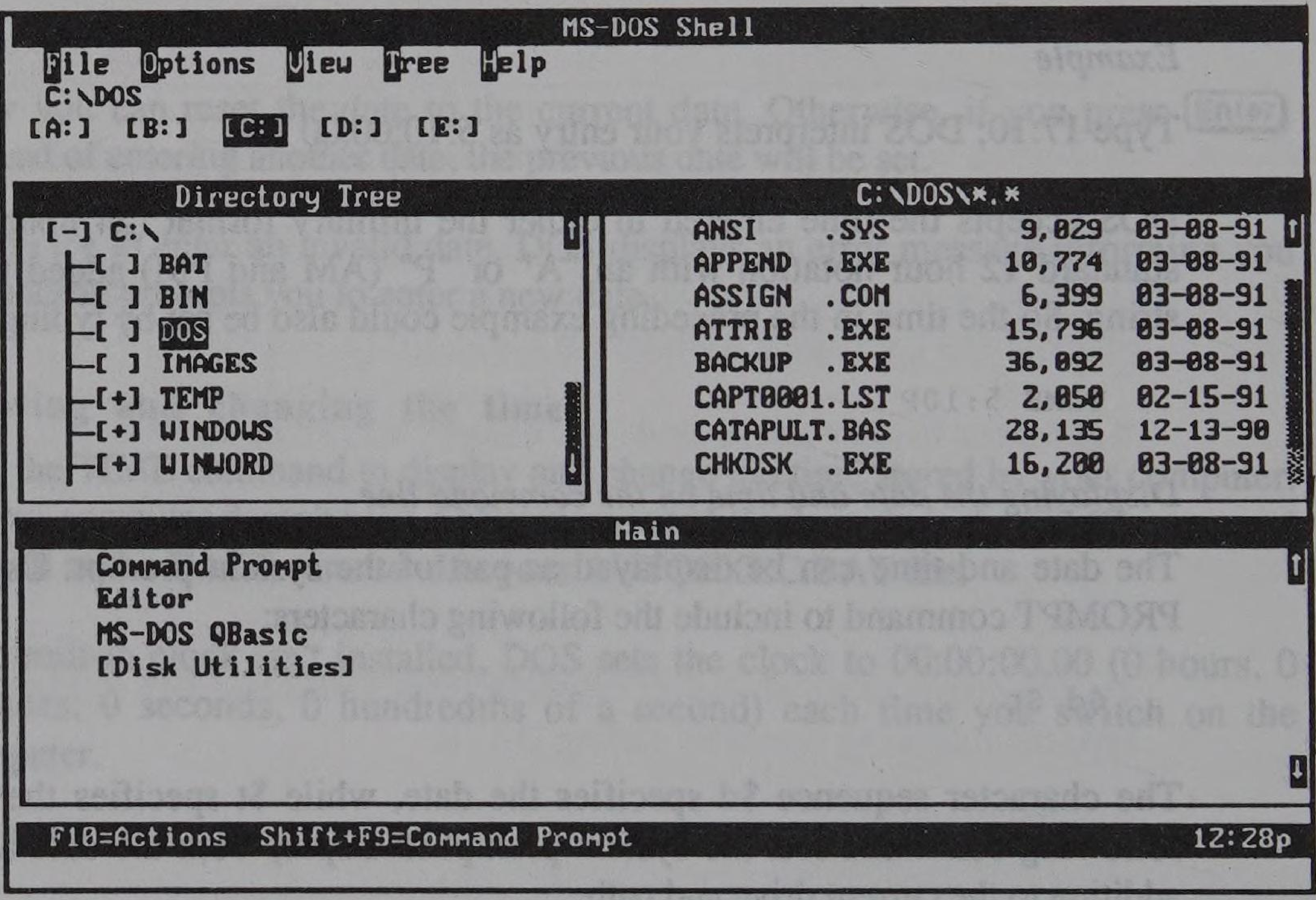
Here is the new prompt created by the previous command:



```
C:\>PROMPT $D $T $P$G

Mon 05-06-1991 11:47:40.37 C:\
```

Displaying the time in the MS-DOS Shell

The MS-DOS Shell displays the time in the lower right corner of the screen. Use the TIME command to make sure the correct time is set.


MS-DOS Shell
File Options View Tree Help
Directory Tree
Main
F10=Actions Shift+F9=Command Prompt
12:28p

```
C:\DOS
[A:]  [B:]  [C:]  [D:]  [E:]

[-] C:\                  ANSI     .SYS     9,029  03-08-91
 ├─[ ] BAT               APPEND   .EXE    10,774  03-08-91
 ├─[ ] BIN               ASSIGN   .COM     6,399  03-08-91
 ├─[ ] DOS               ATTRIB   .EXE    15,796  03-08-91
 ├─[ ] IMAGES            BACKUP   .EXE    36,092  03-08-91
 ├─[+] TEMP              CAPT0001.LST      2,050  02-15-91
 ├─[+] WINDOWS           CATAPULT.BAS     28,135  12-13-90
 ├─[+] WINWORD           CHKDSK   .EXE    16,200  03-08-91
```

C:\DOS\*.*

```
Command Prompt
Editor
MS-DOS QBasic
[Disk Utilities]
```

The time is displayed in the lower-right corner of the MS-DOS Shell

9.2 Customizing The System Prompt

The PROMPT command is used to customize the system prompt.

```
PROMPT Text
```

The Text string contains the parameters which specify how you want the system prompt to appear. We'll explain the parameters below.

Entering PROMPT without parameters resets the prompt to the default setting.

Some characters, which have special meanings, cannot be used with the DOS system prompt. Among these characters are "<" and ">", which are used to redirect input and output, and the pipe symbol "|", which is used for linking several commands.

There are some special codes for the system prompt that DOS doesn't treat like ordinary characters. These codes always start with the special character "$".

$B	Pipe symbol (|).
$D	Display of the current system date.
$E	The ESC character (ASCII 27) used to send commands to ANSI.SYS.
$G	Greater than character ">".
$H	Deletes character to left of cursor (backspace).
$L	Less than character "<".
$N	Displays the current drive.
$P	Displays the current drive and directory.
$Q	Equals sign "=".
$T	Displays the current system time.
$V	Displays DOS version.
$_	New line (CR/LF); permits defining system prompts of more than one line.

Since the dollar sign "$" identifies the special character sequences, it can't be entered by itself. So to use the dollar sign, type two dollar signs:

$$	Represents the "$" character.

Entering PROMPT without any parameters resets the system prompt to NG, which is the default system prompt that displays the drive letter and the greater than character. With this setting, you can use a disk drive as the current drive even if a diskette isn't in the drive, without receiving an error message.

This setting is useful for laptop computers without a hard drive because DOS won't have to access the drive after every command to determine the current directory. Using a simple system prompt without a directory is also useful because the hard disk doesn't have to be accessed unnecessarily, thereby saving valuable battery energy.

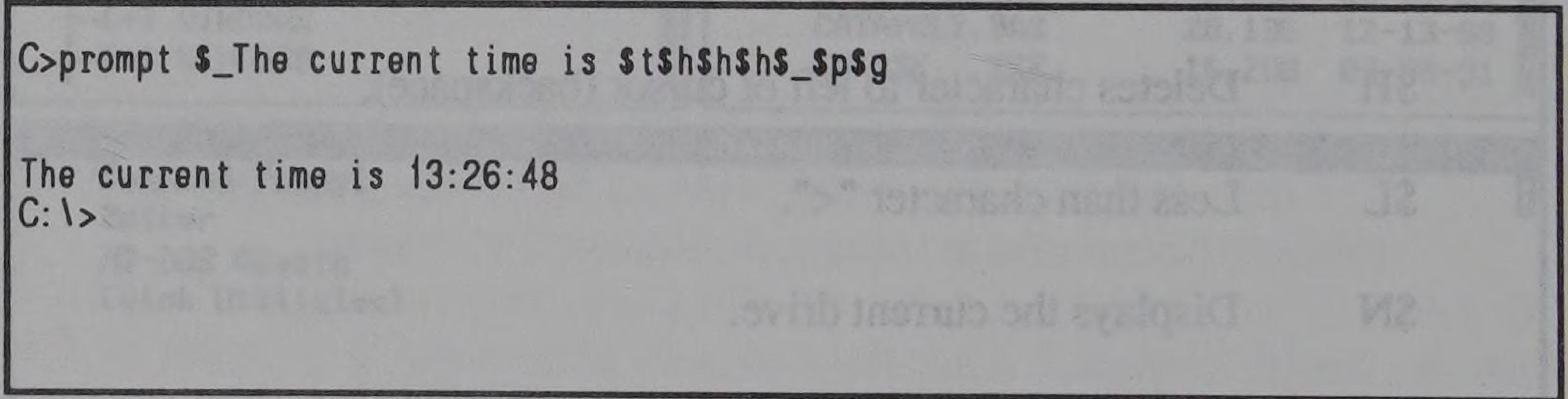
When you're working with subdirectories you should set the system prompt to PG from the AUTOEXEC.BAT file. This causes DOS to display the system prompt with the current drive and directory, followed by the greater than symbol.

Example

To display the current time (without hundredths of a second) and, in the next line, display the current directory as your system prompt, use the following command line:

```
PROMPT $_The current time is $t$h$h$h$_$p$g
```

The trick in this line is to use a backspace ($H) to remove the hundredths of a second.



```
C>prompt $_The current time is $t$h$h$h$_$p$g

The current time is 13:26:48
C:\>
```

Changing the system prompt

The PROMPT command also allows you to enter spaces and displays them on the screen exactly as entered, as shown above.

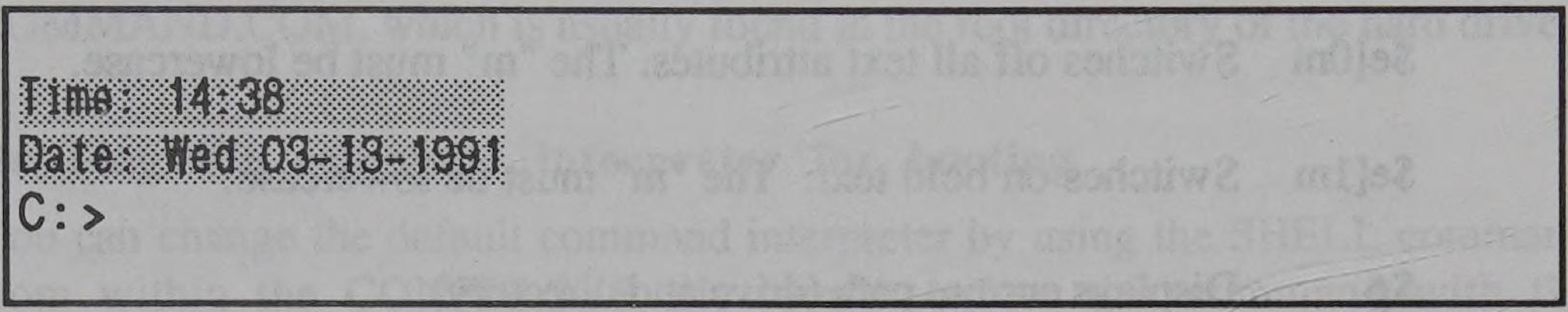
If you load the ANSI.SYS device driver from within your CONFIG.SYS file (enter DEVICE=ANSI.SYS), you can create a more informative header:

```
prompt $e[7mTime: $t$h$h$h$h$h$h            $_Date:
$d$e[0m$e[K$_$e[0m$e[1m$p$g$e[0m
```

This special header always displays the date, time and the current directory.

To use this command line, ANSI.SYS must be installed from your CONFIG.SYS file. Use the following line to install ANSI.SYS:

```
DEVICE=ANSI.SYS
```



```
Time: 14:38
Date: Wed 03-13-1991
C:>
```

Instead of being updated automatically, the date and the time are updated only after you press Enter.

You may have trouble typing the above PROMPT command because the proper ANSI codes must be used and the proper spacing must be maintained in the PROMPT string. So, on the companion diskette we've included a batch file, called HEADMSG.BAT, that can be used instead. This batch file can be edited as needed. You can also change the prompt by calling HEADMSG.BAT from your AUTOEXEC.BAT file. The CONFIG.SYS file must load the ANSI.SYS driver.

To make it easier for you to adapt this system prompt to your own requirements, the following is a brief summary of each code that is used:

$e[7m	This is an escape code for the ANSI.SYS driver. This code switches on the inverse mode for text.
Time:	Displays the text string "Time:".
$t	Displays the current system time.
hhhhhh	Backspaces and deletes six characters to remove the displayed seconds and hundredths of seconds from the screen. Next it displays a string of 9 spaces in reverse video.
$_	Carriage return/line feed.
Date:	Displays the text string "Date:".
$d	Displays the current system date.
$e[0m	Switches off all text attributes. The "m" must be lowercase.

$e[K	Erases all characters from the cursor position to the end of the line. The "K" must be uppercase.
$_	Carriage return/line feed.
$e[0m	Switches off all text attributes. The "m" must be lowercase.
$e[1m	Switches on bold text. The "m" must be lowercase.
$p	Displays current path (drive and directory).
$g	Displays the greater than symbol ">".
$e[0m	Switches off all text attributes. The "m" must be lowercase.

Sometimes when you try to execute the PROMPT command, MS-DOS displays the following message:

```
Out of environment space
```

If this happens, increase the space reserved for environment variables. To do this, specify a higher value for reserved space with the /E switch when opening COMMAND.COM with the SHELL command in the CONFIG.SYS file.

9.3 Specifying The Command Interpreter

The command interpreter accepts and responds to user commands that you type at the keyboard. The standard MS-DOS command interpreter is the file COMMAND.COM, which is usually found in the root directory of the hard drive.

Defining the command interpreter for booting

You can change the default command interpreter by using the SHELL command from within the CONFIG.SYS file (don't confuse this command with the DOSSHELL command).

You may want to use the SHELL command if COMMAND.COM isn't located in the root directory.

```
SHELL=Command_interpreter
```

The drive, path and name for the command interpreter are specified. For example, if the command interpreter is located in the DOS directory of the hard drive, the line in the CONFIG.SYS file would read as follows:

```
SHELL=C:\DOS\COMMAND.COM
```

If the command interpreter is run without any additional parameters, you can leave it at any time by using the EXIT command. However, the computer can no longer accept commands.

Also, a command interpreter started like this doesn't automatically run the AUTOEXEC.BAT file. So basic settings of the AUTOEXEC.BAT file (e.g., search path, keyboard, desired system prompt) aren't available.

You can prevent premature exits from the command interpreter by specifying the /P switch (permanent) after COMMAND.COM. This switch will prevent you from leaving this (the first) command interpreter using the command. The command interpreter also automatically runs the AUTOEXEC.BAT file.

Insert the SHELL command in your CONFIG.SYS file like this:

```
SHELL=C:\DOS\COMMAND.COM /P
```

Here, the CONFIG.SYS file is first run before the commands in your AUTOEXEC.BAT file. The SHELL command loads the command interpreter. Then COMMAND.COM runs the AUTOEXEC.BAT file.

SHELL is only valid in the CONFIG.SYS file. Enter the same data that is specified in COMSPEC (name and path of the command interpreter) after SHELL.

Use the switches following the name of the command interpreter just as you would in a command line. For example, you could define the size of the environment memory for the first command interpreter. DOS stores important settings, such as the search path for programs or the current system prompt, in this environment memory.

Increasing the environment memory

To set the environment memory for COMMAND.COM, in the C:\DOS directory, to 512 bytes, use the following:

```
SHELL=C:\DOS\COMMAND.COM /P /E:512
```

By default, DOS provides only 256 bytes of environment memory.

You should only increase the size of the environment memory area if you get error messages when using SET to create new environment variables or if the changes made with PROMPT and PATH aren't functioning properly. Both of these switches (/P and /E) change the PATH and PROMPT environment variables.

Setting the command interpreter to reload

If an application program requires a lot of memory, DOS will temporarily remove part of the command interpreter from memory. When you exit the application, the command interpreter must be re-loaded.

If you haven't defined the command interpreter search path (with the SHELL command), DOS may ask you to insert the appropriate diskette into drive A:, even if the command interpreter is located on the hard drive. Usually DOS searches for the command interpreter in the root directory of the drive from which the computer is booted.

The command interpreter search path is kept as a system variable. To display the environment variables, type SET. You'll see a statement like this among the system variables:

```
COMSPEC=C:\COMMAND.COM
```

If your command interpreter isn't in the root directory, you'll want to add a line to your AUTOEXEC.BAT file that specifies the location of the command interpreter:

```
SET COMSPEC=C:\DOS\COMMAND.COM
```

The COMSPEC system variable tells DOS that the command interpreter COMMAND.COM is found on the C: drive in the DOS subdirectory. Remember that the name and search path of the COMSPEC environment variable (set by AUTOEXEC.BAT) and SHELL (set by CONFIG.SYS) must match.

If you see one of the following error messages after leaving an application program, an invalid path or filename for the command interpreter was specified. The following appears if the error was in drive A:

```
Invalid COMMAND.COM
Insert COMMAND.COM in drive A:
Press a key to continue...
```

The following appears if the error was in drive C:

```
Invalid COMMAND.COM

Cannot start COMMAND, exiting
```

Press Ctrl + Alt + Del to warm boot your computer and correct the specification for COMSPEC. If this error message appears again, you must boot from a DOS system diskette and correct the settings for SHELL in CONFIG.SYS.

In addition to the search path for reloading the command interpreter, a few other settings are specified in the environment variables. Type SET without any parameters to display the environment variables. The following variables are defined:

PROMPT	Specifies the system prompt.
PATH	Specifies the path specification for executable programs and batch files; DOS searches the directories specified in this path for the program to be executed.
COMSPEC	Specifies the path and name of the command interpreter in case the command interpreter has to be reloaded.

These variables are called "environment variables" because they define the environment in which DOS operates.

If you change any of these variables, eventually you'll notice changes in DOS. The environment variables can be changed as follows:

```
SET Variable=contents
```

To set the system prompt, either use the PROMPT command or change the environment variable directly:

```
SET PROMPT=$P$G
```

Other system variables

The following are some additional system variables that can be set under certain conditions:

APPEND This variable contains the DOS search path for data files (not programs). APPEND is used to define a search path.

The /E switch is used to store a copy of the appended directory list in an environment variable named APPEND. The /E switch can only be used the first time APPEND is called.

TEMP This variable allows you to specify a directory that DOS can use to store temporary variables. If several commands are linked by a filter, the necessary temporary files are stored here.

The MS-DOS Shell also creates some temporary files which are stored in the specified directory. Example: SET TEMP=C:\TMP.

DIRCMD This variable can be used to specify parameters that can be used as defaults for the DOS DIR command. The DIR command uses these parameters as if they were entered directly.

9.4 Using The Command Interpreter

The command interpreter is the heart of DOS. It's loaded automatically every time the computer is switched on and accepts and executes commands that you type at the keyboard. A second interpreter, sometimes called a secondary command interpreter, can also be started like other DOS commands.

Starting a secondary command interpreter

Use COMMAND to start a secondary command interpreter:

```
COMMAND
```

When you change the environment of a secondary command interpreter, the changes only affect the environment temporarily. When the secondary command interpreter terminates, the changed environment is lost and the environment of the first or primary command interpreter is restored. To terminate a secondary command interpreter type:

```
EXIT
```

The original command interpreter, with its unchanged environment, is then reactivated.

Remember not to EXIT from the primary command interpreter. If you do this, you won't be able to send any commands to your computer. To avoid this problem, you can specify the /P switch of the SHELL command to set the command interpreter as permanent.

Example

Many compilers set up directories using special system variables, which are needed for the compiler to function properly. Usually a path to the program files of the compiler must also be defined. Working with these programs and other applications can create conflicts in system variables.

Example

The following sample batch file, STARTUP.BAT, defines the path and directories that are used with the Microsoft C Compiler 5.1:

```
PATH C:\MSC\RBIN;D:\MSC\DOS;D:\MSC;C:\DOS
SET LIB=C:\MSC\LIB
SET INCLUDE=C:\MSC\INCLUDE
SET TMP=C:\TEMP
SET INIT=C:\MSC\SOURCE\ME\INI
```

Suppose the following variables were defined before running the STARTUP.BAT file:

```
PATH=C:\DOS
COMSPEC=C:\COMMAND.COM
PROMPT=$p$g
```

After running STARTUP.BAT, the following environment variables became effective (previously defined variables and new variables). Any changes or deletions must then be made individually:

```
COMSPEC=C:\COMMAND.COM
PROMPT=$P$G
PATH=C:\MSC\RBIN;C:\MSC\DOS;C:\DOS
LIB=C:\MSC\LIB
INCLUDE=C:\MSC\INCLUDE
TMP=C:\TEMP
INIT=C:\MSC\SOURCE\ME\INI
```

It's easier to start a new command interpreter before calling STARTUP.BAT and leave it, after working with the compiler, by using EXIT. This method automatically restores the previous environment values:

```
COMMAND
STARTUP
```

Here, all of the new environment variables required by the compiler are defined. After working with the compiler, restore the original environment by typing:

```
EXIT
```

Many application programs let you execute one or more DOS commands from within the application program itself. To do this, a command interpreter is run by the application. Usually you can return to the application program by typing EXIT.

COMMAND.COM can be called from within an application in the same way that it is called from the system prompt. After entering EXIT, all the changed environment variables for the command interpreter are restored to their original values.

When a new command interpreter is started from within an application program, there is usually less memory available for programs. At least a part of the application program is occupying memory.

For example, if 550K of memory is available after booting your computer, after starting Microsoft Word 5.5 from DOS only 200K - 250K will be available. This means that only relatively small programs can be run. If a large program is called, an error message will appear.

Starting a command interpreter with a larger environment

If the environment memory isn't large enough, DOS displays an error message when you try to change or set up an environment variable.

```
Out of environment space
```

A new command interpreter can be started with a larger area reserved for environment memory:

```
COMMAND /E:xxxxx
```

For xxxxx, specify the size of the environment memory in bytes (256 default). Values between 160 and 32,768 bytes are acceptable.

When this command interpreter is terminated by the EXIT command, the original setting for the environment will be effective again.

To permanently change the size of the memory area reserved for the environment variables, insert a line in the CONFIG.SYS file using the SHELL command:

```
SHELL=COMMAND.COM /E:xxxxx
```

Enter the correct path to the command interpreter and choose an appropriate environment size, in bytes, for xxxxx (512 bytes would be appropriate).

Using the command interpreter for a single command

The /C switch can be used to load a command interpreter and run a single command:

```
COMMAND /C Command_line
```

Enter the commands and parameters in place of command_line. DOS starts a new command interpreter, executes the command line and automatically stops and terminates the new, temporary COMMAND.COM command interpreter.

This method is useful for running batch files that change the environment. This is especially useful when the changes need to be cancelled immediately and the default environment restored.

The /C and /P switches cannot be combined. /C loads the command interpreter for a single command and /P loads COMMAND.COM in memory permanently (can't exit).

9.5 Environment Variables

Each command interpreter contains an environment in a special area of memory. Important information about the working environment and system settings are stored there. Also, user defined variables can be defined in this area.

An environment variable is a basic setting of DOS that can be changed. There are many settings in DOS that either can't be changed or that can only be changed through very complicated procedures. However, the settings stored in environment variables can be adjusted easily by simply changing the variables.

Some variables change the way the system works (system variables). Other variables don't have any direct effect, but can be used by batch files (user defined variables). When using variables in batch files, enclose the variable name within percentage signs.

Displaying environment variables

Use SET to display the existing variables. The individual variables are displayed as:

```
Variable_name=value
```

Depending on the AUTOEXEC.BAT and CONFIG.SYS, most of the following variables are defined and displayed after entering SET:

```
COMSPEC=C:\COMMAND.COM
PATH=C:\DOS
PROMPT=$p$g
```

The first line specifies the path and filename of the command interpreter in case it must be reloaded after running an application. (For example, after exiting from Windows, COMMAND.COM must be reloaded into memory again.) The second line contains the search path for commands and batch files; use PATH to change this line. The third line contains the specification for the system prompt, which can also be changed as explained earlier in this chapter.

Defining and changing environment variables

New variables can be defined and existing variables can be changed by using the SET command.

```
SET Variable_name=value
```

If no values are specified after a variable in the SET command, the existing variable is deleted.

By default, DOS reserves only 256 bytes of memory for environment variables. So you may see the following error message if you're defining or changing several environment variables:

```
Out of environment space
```

One solution to this problem is to start a new command interpreter with a larger environment (/E switch) or change the SHELL statement in the CONFIG.SYS file by adding the /E switch.

SET is also used to change the environment variables for the search path and the system prompt. Using SET has the same effect as changing the variables with PATH and PROMPT.

Example

To set a new search path for commands and batch files to look in the C:\DOS and C:\WORD directories, use the following:

```
SET PATH=C:\DOS;C:\WORD
```

Only changes in the system variables PATH, APPEND, COMSPEC and PROMPT and the TEMP and DIRCMD variables change the way DOS works. The user defined environment variables don't have any effect on DOS. However, batch files can read and use user defined environment variables.

For example, you can write a batch file that backs up a directory, specified in the environment variable BACKUP, onto diskettes. Use SET to define these environment variables before calling the batch file.

The environment can't be changed from special user interfaces, such as the Norton Commander®. These types of programs use a copy of the environment. When you exit the program, the original environment is restored.

Special environment variables

The following is a list of several system variables:

PATH	Contains the search path for programs and batch files; DOS looks for programs in the current directory first, then in all the directories specified in the search path. The search path should contain at least the directory for DOS commands.

APPEND	Contains the current search path for data files defined by the APPEND command. If APPEND hasn't been used or the /E switch wasn't used with the APPEND command, the variable is not defined.
PROMPT	Contains the specifications for the system prompt for the command line.
COMSPEC	Contains the path and name of the command interpreter in case it must be reloaded after running an application. Some programs also use the system variable COMSPEC to execute DOS commands.
TEMP	Specifies the directory DOS uses to store temporary files. If several commands are linked using filters, the necessary temporary files are stored in the directory specified here. The MS-DOS Shell also creates some temporary files that are stored in this directory.
DIRCMD	Contains the default parameters for the DOS DIR command. The DIR command handles these parameters as if they were entered directly from the keyboard. For example, you can use SET DIRCMD=/P to display the diskette directory one screen page at a time. You should define this environment variable from the AUTOEXEC.BAT file.

The environment variables mentioned above have a special meaning to DOS. Many applications also use environment variables. For example, Word 5.0 sets up temporary files in the WORD directory and in the current directory. If a directory is specified in an environment variable (TMP), WORD stores temporary files in this directory. The user manuals for application programs provide information about possible environment variables.

Unfortunately, some programs (Word 5.0, dBASE IV) use TMP as the variable for the temporary directory, while DOS 5.0 uses the TEMP variable. Both variables can be defined to use the same directory.

9.6 Special System Settings

This section describes three special commands that you can specify to DOS. One defines whether the <Break> key can be used to interrupt or stop commands and batch files during execution. Another specifies whether disk write operations should be verified. The third specifies whether commands should be displayed on the screen.

Switching the break on/off

You can use the <Break> key, or Ctrl + C (or also Ctrl + Pause), to interrupt commands when DOS is reading a character oriented device (keyboard) or displaying data (printer).

To use the <Break> key during read or write accesses to hard drives and diskettes (block oriented input/output), the BREAK command must be switched on with the following command:

```
BREAK ON
```

You can switch it off by typing:

```
BREAK OFF
```

Entering BREAK without any other parameters displays the current status. The default setting is OFF.

Usually the break option should be switched on. For example, if you accidentally copy an extensive file, you can abort this process by pressing Ctrl + C if BREAK is switched on. Many compilers require extensive file operations and can only be interrupted when BREAK is switched on.

However, Ctrl + C can't be used to interrupt all applications. For example, Microsoft Word 5.0 ignores the <Break> key.

Switching VERIFY on/off

Use the VERIFY command to specify whether or not DOS should check the data after writing it to your hard drive or diskette:

```
VERIFY ON
```

This command line switches on VERIFY. Unfortunately verifying each write access of a device slows down the system. Switch off this option by typing:

```
VERIFY OFF
```

To display the current status on the screen, enter VERIFY without any other parameters. The default setting is OFF.

Switching off ECHO

Use the ECHO command to specify whether or not DOS command lines are displayed from within batch files as they're executed:

```
ECHO ON
```

Enter ECHO ON to display command lines. This option can be switched off by typing:

```
ECHO OFF
```

If ECHO is switched off at the command line, remember that it must be switched on again. If ECHO isn't switched on, the system prompt will no longer be displayed. Entering ECHO without any other parameters displays the current status. The default is ON.

For example, you can easily change the status within the batch file by placing ECHO OFF at the beginning and ECHO ON at the end of a batch file. By doing this, only the messages of the commands that are being processed in the batch file will be displayed on the screen.

This is less complicated than displaying every command line before processing it. However, displaying all the command lines can be helpful when you're trying to find errors in a batch file.

If ECHO is switched off in the first line of the batch file, this line will still be displayed. If you don't want this line displayed, place a "@" symbol in front of the line. This symbol will prevent the line from being displayed.

Remember that ECHO is also used for displaying additional text messages in batch files. So this command has a double meaning.

If ECHO is switched off, you cannot use PROMPT to send any control sequences for the screen or the keyboard to ANSI.SYS. If you want to use this option in batch files, switch on ECHO before the PROMPT instruction and switch it off afterwards.

9.7 Setting The Number Of Buffers And Files

The number of read/write buffers and the maximum number of files that can be open simultaneously can be set in the CONFIG.SYS file. By selecting the proper values for these settings, you can increase the performance of DOS. Remember that the changes made to any statement in the CONFIG.SYS file won't take effect until the computer is rebooted.

Setting the read/write buffer

Accessing the data on disk drives and hard drives is time-consuming. The more a drive needs to be accessed, the slower your computer operates. Rather than reading or writing individual bytes of data, the computer works with blocks of information. This information is collected in buffers.

To set the number of buffers, include the following line in the CONFIG.SYS system file:

```
BUFFERS=XX
```

For XX, specify the number of buffers needed. Each of these buffers uses 528 bytes of memory. Of this total, 512 bytes are used to store the data and 16 bytes are used to manage that data. You should specify between 15 and 20 buffers. However, if the directory structure on a hard drive is large, you can specify a larger number of buffers.

If you don't explicitly have a BUFFER statement in the CONFIG.SYS file, DOS reserves a number based on how much memory is available. DOS uses 10 buffers as the default setting for memory between 256K and 511K. If more than 512K of memory is available, DOS reserves 15 buffers.

If you use a cache program, such as SMARTDRV (which is part of MS-DOS 5.0), to speed up access to your hard drive, you must decrease the number of buffers set with BUFFERS. Otherwise, you will not only waste memory, but also reduce the speed of your computer.

Many installation programs for applications change the value of BUFFERS in CONFIG.SYS if the file doesn't contain the desired minimum value. After installing a program, check whether such a change is really desirable. For example, if you use an acceleration program, such as SMARTDRV.SYS, usually it isn't necessary to increase the value for BUFFERS.

Number of files opened at the same time

In order for DOS to obtain information from a file or write to it, the file must be open. DOS requires a minimum of five open files and uses 7 as a default value. Using the default, your applications would only be able to open two files. This usually isn't enough for most applications. Many applications require a series of files to be opened simultaneously. For example, Microsoft Word 5.0 requires the text file, the style sheet, any glossaries and temporary files to be open at the same time.

To set the number of opened files, use the following line in the CONFIG.SYS:

```
FILES=XX
```

For XX, enter the number of files you want opened simultaneously. You should specify a value between 20 and 30. Each additional file that's specified requires an additional 48 bytes of memory.

During installation, many application programs check the number specified after FILES and, if necessary, increase it to a larger value. Don't decrease this value; otherwise the application may not function properly.

Chapter 10

Peripheral Devices

DOS can be adapted to your existing hardware in various ways. The most important ways to adapt DOS to your hardware include changing the way it works with the printer, configuring the screen and changing the capacity of disk drives.

DOS works with devices through *driver* programs, which manage the devices. Some of these drivers are part of DOS (drivers for disk drives and the keyboard), while others can be specified in the CONFIG.SYS file. The statements in the CONFIG.SYS file are processed as the computer is started.

10.1 Installing A Printer

The MODE command can be used to set the mode of operation of a printer to the parallel port or to redirect the printer output to a serial port. These settings can be changed at any time.

Setting the printer output

The number of characters per line and the line spacing for a parallel printer can be specified.

> These settings only work on IBM or EPSON compatible dot-matrix printers. Many of the setting options can only be used under DOS because applications usually use their own special programs for printers (printer drivers). These programs either reset the DOS settings or completely ignore them.

Setting the number of characters per line

With the MODE command you can increase the number of characters printed in each line from 80 to 132. However, your printer must support a font that is narrow enough to do this. To change the characters per line for a printer connected to the first printer port (LPT1), use the following command:

```
MODE LPT1:132
```

The following command resets the number of characters printed in each line to normal font size (i.e., 80 characters per line):

```
MODE LPT1:80
```

Setting line spacing

MODE can also be used to change line spacing. Type the number of lines per inch after the number of characters:

```
MODE LPT1:Number_of_characters,X
```

For X, specify the value 6 or 8. If you want to change the number of lines without changing the number of characters per line, place a comma in front of the number of lines:

```
MODE LPT1:,X
```

The default setting is 80 characters and 6 lines per inch.

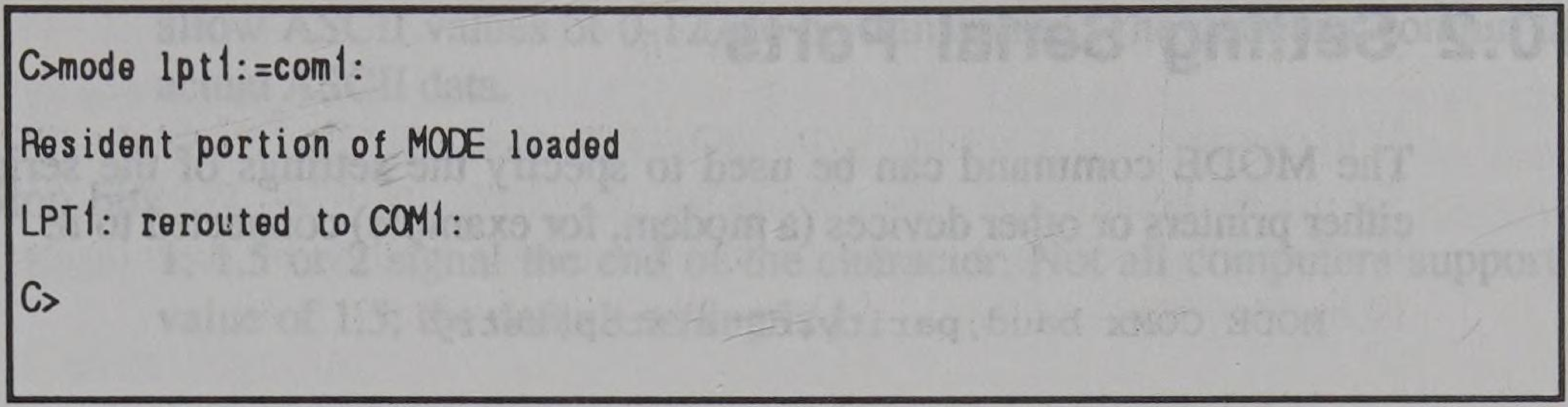
Many applications (especially word processors) reset the printer. So settings made with MODE are no longer valid.

Redirecting printer output

Data can be redirected from a parallel port to a serial port. Occasionally this must be done if a laser printer can only be connected to the serial port but an application always prints to the first parallel port (LPT1):

```
MODE LPTx=COMy
```

After you specify the number of the parallel port for x and the number of the serial port for y, you'll see a message similar to this.



```
C>mode lpt1:=com1:

Resident portion of MODE loaded

LPT1: rerouted to COM1:

C>
```

Printer LPT1: has been redirected to serial port 1

MODE can also be used to cancel the redirection by typing:

```
MODE LPTx
```

Before redirecting printer output to the serial port, make the proper settings on the serial port, as explained in the following section.

10.2 Setting Serial Ports

The MODE command can be used to specify the settings of the serial port for either printers or other devices (a modem, for example) connected to it:

```
MODE COMx baud,parity,data,stop,retry
```

Baud Sets a baud rate between 110 and 19200 baud. The following are possible values that can be used for baud:

Setting	Baud Rate
11	110
15	150
30	300
60	600
12	1200
24	2400
48	4800
96	9600
19	19200

The baud rate specifies the speed of data transfer in bits per second. As shown in the table above, this is a two digit value that specifies the transmission rate in bits per second. To convert this speed (baud rate) to characters (bytes) per second, divide the value by 10. For example, a baud rate of 300 will send approximately 30 characters per second (30 * 8 data bits, 1 parity bit and 1 stop bit).

Parity Parity is a method to detect data transmission errors. The value can be any of the following:

n	no parity
e	even parity
o	odd parity
m	mark
s	space

The default value is e. Since mark (m) and space (s) refer to TTY printers, they may not be applicable to all computers.

Data bits

This value specifies the number of data bits in a character and may contain a value between 5 and 8. The default value is 7. Not all computers will support a value of 5 or 6. When 7 data bits are specified, 2 stop bits can be selected, as explained below. However, 7 data bits only allow ASCII values of 0-127 to be transferred. The data bits contain the actual ASCII data.

Stop bits

1, 1.5 or 2 signal the end of the character. Not all computers support a value of 1.5; the default setting is 1.

Retry Specifies the retry action to be taken if a timeout error occurs when attempting to send data to a parallel printer. Using this parameter causes a portion of the MODE command to remain in memory. The following table shows the possible values for retry and a brief description:

e	Return an error from a status check of a busy port.
b	Return "busy" from a status check of a busy port.
p	Continue trying to send data until the printer accepts output.
r	Return "ready" from a status check of a busy port.
n	No retry action. This is the default value.

Don't use any values for retry if you're using a network.

If a comma is used instead of a parameter setting, the most recent settings are used for the omitted parameter.

Examples

The following is an example of how to specify these settings:

```
MODE COM1: 300,n,8,1
```

The baud rate must be specified but all the other parameters are optional. The default setting for the other values are: e,7,1, which means even, 7 data bits and 1 stop bit. A comma can be used for parameters. (Assuming that MODE hasn't been used to change the other parameters previous to this call.)

The following is an example with only the baud rate:

```
MODE COM1:1200
```

This call automatically sets the default parameters to even parity, 7 data bits and 1 stop bit along with 1200 baud.

To set the number of data bits to 8 with the same baud rate, use the following command:

```
MODE COM1:1200,,8
```

10.3 Setting The Display Mode

You can also use a variation of the MODE command to specify the display mode. To use this command, your computer must have an EGA, VGA, CGA or another type of graphics card. You cannot change the settings on computers that use a Hercules Graphics Card.

```
MODE display_adapter
```

Setting the display mode

The following parameters can be used for the mode of display:

mono	Monochrome display, 80 characters per line
40	40 characters per line, color graphics adapter (CGA, EGA, VGA)
80	80 characters per line, color graphics adapter
co40	40 characters per line, color display active, color graphics adapter
co80	80 characters per line, color display active, color graphics adapter
bw40	40 characters per line, color display not active, color graphics adapter
bw80	80 characters per line, color display not active, color graphics adapter

You can only set a mode that is appropriate for the type of graphics card installed in the computer. MODE can also be used to switch between several installed graphics cards. For example, you may do this if you connected more than one monitor for debugging.

Changing the settings for the display mode is especially useful when an application changes the mode but doesn't reset the original mode after you exit the program. For example, if you run dBASE IV on a computer with an EGA card with 43 lines per screen, the computer retains this setting even after you quit dBASE IV. To return to the usual mode of 25 lines per screen, type:

```
MODE CO80
```

If you find it difficult to work with this number of lines per screen, install the ANSI.SYS device driver in the CONFIG.SYS file using the /L switch.

If you try to set a mode that the graphics card doesn't support, an error message similar to the one shown below will be displayed.

```
Function not supported by this computer
```

Setting the number of text lines in the display

You can change the number of lines that DOS displays on the screen if your computer has an EGA or VGA graphics card. On a computer with an EGA card, use the following command to set an 80 character color display with 43 lines per screen:

```
MODE CO80,43
```

If the graphics card cannot display the specified number of lines, the following error message appears:

```
Function not supported by this computer
```

When changing the number of displayed text lines, your computer requires the extended screen control system. This means that the ANSI.SYS device driver must be installed by the CONFIG.SYS file. If ANSI.SYS isn't installed, DOS will display a message similar to the following:

```
ANSI.SYS must be installed to perform requested function
```

Changing lines and columns in the display

With EGA and VGA graphics cards, you can also change the number of lines that DOS displays without setting the screen mode. Use the command words COLS (columns) and LINES (lines).

The following syntax is used:

```
MODE CON Command_word=Numerical_value
```

Depending on the capacity of the graphics card, either the value 40 or 80 can be used for COLS and the value 25, 43 or 50 can be used for LINES. For a VGA card, use the following command line to set the maximum number of lines in DOS for the screen display:

```
MODE CON LINES=50
```

The ANSI.SYS device driver must be installed by the CONFIG.SYS file.

10.4 Screen Control System And Keyboard Actions

DOS addresses the screen and keyboard under the common device name of CON. When writing, CON corresponds to the screen; when reading, CON corresponds to the keyboard.

Preparing for screen control and keyboard actions

The device driver ANSI.SYS lets you use special commands for the screen and the keyboard. The CONFIG.SYS file must contain the following command line:

```
DEVICE=ANSI.SYS
```

If the file ANSI.SYS isn't in the root directory, its complete path must be specified. Any new options provided by ANSI.SYS aren't available until you restart the computer.

Since this device driver requires memory, use ANSI.SYS only when absolutely necessary. The memory ANSI.SYS uses will no longer be available to applications.

Sending commands to ANSI.SYS

To use these features of ANSI.SYS, you must send it commands that it understands. You do this by preceding the command with an "escape sequence".

The easiest way to do this is to use a special character sequence ($e[) following the PROMPT command.

DOS automatically converts this to correct escape sequences and passes the commands to ANSI.SYS.

Screen control system with ANSI.SYS

Using ANSI.SYS to extend the screen control system provides many possibilities. For example, you can set the colors for the foreground and the background of the screen as well as change the way characters are displayed. These changes are only valid with DOS. Many applications change these settings when they're activated.

Changing the character display

Use "m" to change the way characters are displayed. The syntax for this change is as follows:

```
PROMPT $e[xm
```

"m" is the actual command for changing the display while "x" represents the desired character attribute:

Value for x	Effect on the character display
0	normal
1	bold
4	underlined (with a color graphics card: change in color)
5	blinking
7	inverted display
8	invisible

Example

To switch the display for all of the subsequent characters to bold font, use the command:

```
PROMPT $e[1m
```

To switch the display back to default font at any time, simply type:

```
PROMPT $e[0m
```

To create the brackets "[" and "]" on international keyboards, press Alt + 9 1 and Alt + 9 3 on the numeric keypad.

These escape sequences only work if ANSI.SYS has been installed by CONFIG.SYS prior to starting your computer.

The control sequences are sent to ANSI.SYS each time the prompt is displayed. So, if you use the PROMPT line to set bold type, the command sequence is sent each time the prompt is displayed, even though the change has already been made. In many cases this isn't necessary because switching to another text attribute once is sufficient.

In this type of situation it's better to use two separate command lines. The first line contains the control sequence to switch the text attribute and the second line switches on the original prompt message. In the example below, the first line switches on bold text and the second line defines the system prompt as the path and directory, followed by the greater than symbol (>).

```
PROMPT $e[1m
PROMPT $P$G
```

Changing the color of the font

If you have a color monitor and a color graphics adapter, the color of the displayed text can also be changed. Use the same syntax that is used to change the character display but enter a value between 30 and 37 for "x":

Value for x	Effect on the font color
30	black
31	red
32	green
33	yellow
34	blue
35	magenta (violet)
36	cyan (light blue)
37	white

Changing the color of the screen background

To change the color of the screen background, specify a value from the following table for "x" (only for color graphics):

Value for x	Effect on the background color
40	black
41	red
42	green
43	yellow
44	blue
45	magenta (violet)
46	cyan (light blue)
47	white

Samples of font and background color

In order to make several changes to the screen display, separate the individual changes with a semicolon. However, don't place a semicolon in front of the actual command character "m".

The following command changes the font color to red and the background color to blue on computers with a color screen:

```
PROMPT $e[31;44m$P$G
```

To increase the intensity of the displayed characters, change the command to:

```
PROMPT $e[1;31;44m$P$G
```

Changing the definition of the screen display

As we previously explained, the DOS MODE command can be used to set the screen display. Another way to change the display mode is by sending special command sequences to ANSI.SYS. Use the lowercase character "h" as the command code. The syntax for the command is as follows:

```
PROMPT $e[xh
```

"h" is the actual command for changing the display, while "x" sets the screen display according to the following table:

Value for x	Screen display
0	40*25 monochrome
1	40*25 color
2	80*25 monochrome
3	80*25 color
4	320*200 color
5	320*200 monochrome
6	640*200 monochrome
7	Turns on automatic word wrapping. Use 7l to turn it off. If word wrapping is turned off, new characters that you enter overwrite the last character in the screen line.
14	640*200 color
15	640*350 monochrome
16	640*350 color
17	640*480 monochrome
18	640*480 color
19	320*200 color

Example

Use the following commands to set a graphics screen to display 640 * 480 pixels:

```
PROMPT $e[18h
PROMPT $p$g
```

Special features when using batch files

ANSI.SYS receives the control sequences from PROMPT only when ECHO is switched on and at least one system prompt is displayed on the screen. This is especially important for batch files which frequently switch off ECHO.

Example

The following batch file is incorrect and doesn't change the font because ECHO was switched off while the command line "PROMPT $e[1m" was being processed.

```
@ECHO OFF
PROMPT $e[1m
PROMPT $P$G
ECHO ON
```

While ECHO is switched on, the correct system prompt must be displayed with the ESC command for at least one command line. The command line can be blank; so no text is displayed on the screen. The following is the correct version of the batch file, this batch file is contained on the companion diskette in the BAT directory:

```
@REM FONT.BAT Chapter 10
@ECHO OFF
REM ECHO is turned off here
REM These lines are not displayed
REM "ECHO" is turned on now, but
REM the display of the lines is prevented by "@":
@ECHO ON
@REM Now we are sending the command line, the line itself
@REM isn't displayed yet:
@PROMPT $e[1m

@REM Turn ECHO off again
@ECHO OFF
REM The command was sent to ANSI and the font
REM is switched
PROMPT $P$G
ECHO ON
```

The blank line is important. For each blank line, a system prompt is displayed on the screen. However, since the system prompt consists of an ESC sequence, only a blank line is displayed.

Using ANSI.SYS to program the keyboard

To assign a new character to a key, specify the ASCII code of the key that should be changed after the escape code for the command "$e[". Enter a semicolon followed by the ASCII value of the new key. Then enter "p", which is the command for changing the keyboard layout.

Unlike DOS commands, you must enter the parameter first and then the letter that identifies the command when using the ANSI.SYS driver.

Example: Switching Y and Z

If you want to play a joke on a friend, enter the following two lines on his/her computer when he/she isn't looking (ANSI.SYS must have been installed by CONFIG.SYS):

```
PROMPT $e[121;122p
PROMPT $e[122;121p
```

The first line changes "y" to "z" and the second line changes "z" to "y". Your friend will be thoroughly confused when he/she realizes that this only applies to lowercase letters; the uppercase letters aren't changed.

You need to define a key to ANSI.SYS only once. Set the system prompt back to the original value. After entering the first two PROMPT lines, the system prompt will no longer appear.

```
PROMPT $P$G
```

Cancelling the keyboard definition

To undo this definition, simply type the same command line without the semicolon and without the second ASCII value:

```
PROMPT $e[121p
PROMPT $e[122p
```

The "y" and "z" keys will work normally again.

Changes to the keyboard layout aren't valid in all programs. Many applications, such as word processors and spreadsheets, have their own defined keyboard layouts. As soon as you exit the application, the redefined keyboard for DOS will be activated again.

Layout of special keys

Programming special keys is more useful than redefining letter keys. Special keys include the function keys and the combination keys Alt, Shift and Ctrl. Since these keys are seldom used, you can safely assign new characters or character sequences to them.

These special keys have number codes that are specified in the definition by a zero and a semicolon in front of the numeral (0;x). For example, to redefine the F2 key to type the backslash "\" character, use the following command:

```
PROMPT $e[0;60;92p
```

When you change the keyboard layout with ANSI.SYS, the change isn't necessarily valid in all applications. For example, Word 5.0 ignores function keys changed with ANSI.SYS. Word has its own options for defining macros, character strings or commands in function keys.

Using EDLIN to create command files

If you don't like changing the keyboard layout by using PROMPT lines, you can use EDLIN instead. EDLIN can be used to create the files containing the necessary commands for ANSI.SYS. However, this can only be done with EDLIN; word processors such as Word 5.0 or the MS-DOS editor EDIT cannot be used.

In EDLIN, enter the ESC character by typing Ctrl + V and then the bracket [. To type the control sequence ESC[, press Ctrl + V [[(press the bracket key twice).

For more information about using EDLIN, refer to Chapter 12.

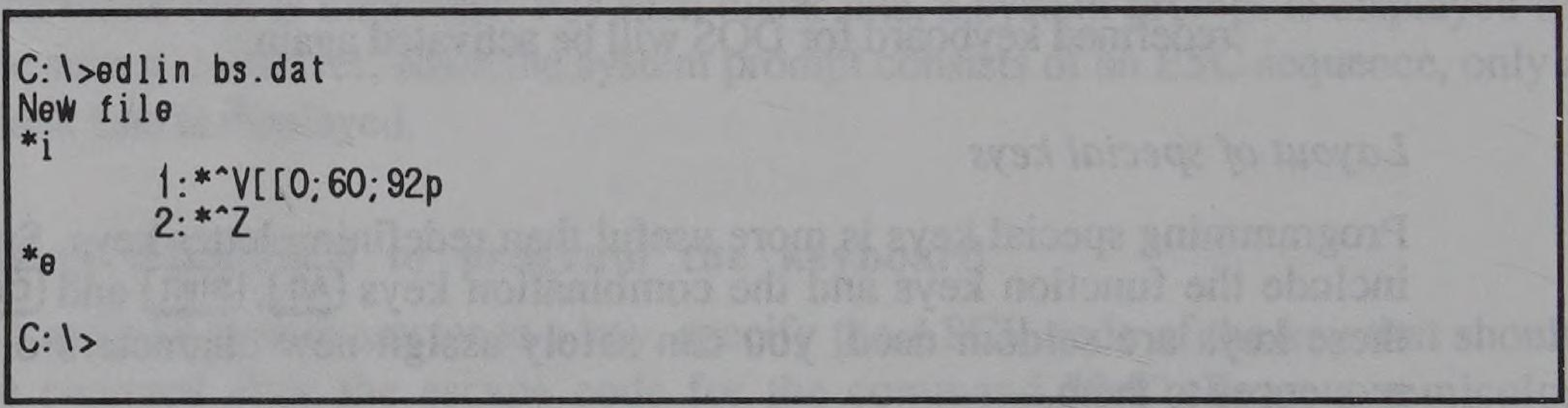
EDLIN is used to create the file BS.DAT. Start EDLIN by typing:

```
EDLIN BS.DAT
```

A line containing the message "New File" is displayed. Then a new line, in which the normal system prompt is replaced with an asterisk, is displayed. Enter a I to insert lines in the new file. Enter "Ctrl V [[0;60;92p" for the command sequence. The line in EDLIN should look like the following ("1:*" represents the current line 1):

```
1:*^V[[0;60;92p
```

Press Enter to end the line and enter Ctrl + Z to leave insert mode. Now press E and Enter to save the file and quit EDLIN.



```
C:\>edlin bs.dat
New file
*i
        1:*^V[[0;60;92p
        2:*^Z
*e

C:\>
```

The BS.DAT file using EDLIN

To activate this change in the keyboard, the file must be output to the screen. Type:

```
TYPE BS.DAT
```

The screen will display a blank line as ANSI.SYS processes the control sequence. When you press F2, the backslash appears on the screen. You can make more changes to the keyboard by entering the command lines in EDLIN. The change is completed when the file is displayed on the screen with TYPE.

Modifying the keyboard with commands and automatic execution

An entire sequence of characters can be assigned to a key. The character sequence must be enclosed in quotation marks. For example, to assign the character sequence "DIR /W" to F1 (code 59), type the following PROMPT line:

```
PROMPT $e[0;59;"DIR /W"p$P$G
```

Press F1; the command appears after the system prompt. To execute the command without having to press Enter, add the ASCII code 13 (Enter) to the command:

```
PROMPT $e[0;59;"DIR /W";13p$P$G
```

Command shortcuts

After you've assigned DOS commands to all of the function keys, you may not be able to remember which key belongs to which command. So you should use the first letter of a command in conjunction with the Alt key. For example, to assign the FORMAT A: command to Alt + F, type the following command line:

```
PROMPT $e[0;33;"FORMAT A:"p
```

Commands that could destroy data shouldn't be executed automatically. So don't add ASCII code 13 (Enter) to the command line.

In Chapter 17 you'll find additional information on reassigning keys.

Key codes

While the ASCII code can be specified for "normal" keys, special values must be specified for extended keys. These values always start with a zero, followed by a semicolon and the second value. The following table shows these values:

Key	Code	Key	Code
F1	59	Alt+1	120
F2	60	Alt+2	121
F3	61	Alt+3	122
F4	62	Alt+4	123
F5	63	Alt+5	124
F6	64	Alt+6	125
F7	65	Alt+7	126
F8	66	Alt+8	127
F9	67	Alt+9	128
F10	68	Alt+0	129
F11	133	Alt+-	130
F12	134	Alt+F1	104
Ins	82	Alt+F2	105
Del	83	Alt+F3	106
Home	71	Alt+F4	107
End	79	Alt+F5	108
PgUp	73	Alt+F6	109
PgDn	81	Alt+F7	110
Alt+A	30	Alt+F8	111

Key	Code	Key	Code
Alt+B	48	Alt+F9	112
Alt+C	46	Alt+F10	113
Alt+D	32	Alt+F11	139
Alt+E	18	Alt+F12	140
Alt+F	33	Shift+F1	84
Alt+G	34	Shift+F2	85
Alt+H	35	Shift+F3	86
Alt+I	23	Shift+F4	87
Alt+J	36	Shift+F5	88
Alt+K	37	Shift+F6	89
Alt+L	38	Shift+F7	90
Alt+M	50	Shift+F8	91
Alt+N	49	Shift+F9	92
Alt+O	24	Shift+F10	93
Alt+P	25	Shift+F11	135
Alt+Q	16	Shift+F12	136
Alt+R	19	Shift+Tab	15
Alt+S	31	Ctrl+F1	94
Alt+T	20	Ctrl+F2	95
Alt+U	22	Ctrl+F3	96
Alt+V	47	Ctrl+F4	97
Alt+W	17	Ctrl+F5	98
Alt+X	45	Ctrl+F6	99
Alt+Y	44	Ctrl+F7	100
Alt+Z	21	Ctrl+F8	101
←	75	Ctrl+F9	102
→	77	Ctrl+F10	103
↑	72	Ctrl+F11	137
↓	80	Ctrl+F12	138
Ctrl+Prt Sc	114	Ctrl+PgUp	132
Ctrl+←	115	Ctrl+PgDn	118
Ctrl+→	116	Ctrl+End	117
Ctrl+Home	119		

Beginning with DOS 4.0, ANSI.SYS supported extended keyboards (with extra keys such as F11 and F12). These extended keys can be accessed by specifying the /X switch when calling ANSI.SYS:

```
DEVICE=ANSI.SYS /x
```

10.5 Modifying The Keyboard

You can change the settings for the character set and speed of the keyboard.

Selecting the keyboard

The MS-DOS default keyboard is the American keyboard. To install a foreign language keyboard, such as the British keyboard, use the following command line:

```
KEYB UK
```

The abbreviation "UK" sets the British (United Kingdom) keyboard definition. Normally KEYB searches in the current directory for the file with the keyboard definitions named KEYBOARD.SYS. So if the file isn't in the root or current directory, specify the complete path.

```
KEYB UK,,C:\DOS\KEYBOARD.SYS
```

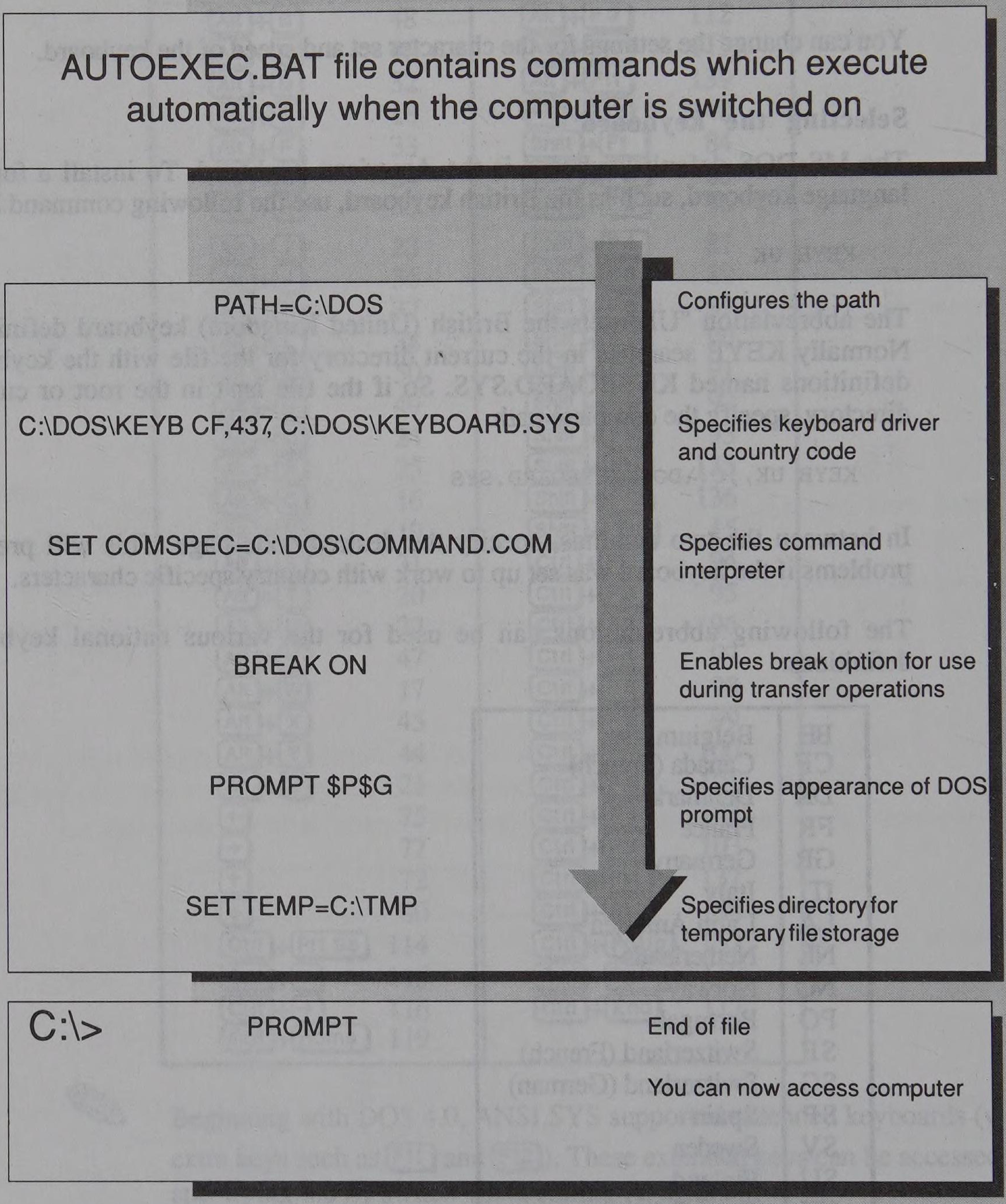
In between the two commas, specify the desired code page. This will prevent problems if the keyboard was set up to work with country specific characters.

The following abbreviations can be used for the various national keyboard definitions:

BE	Belgium
CF	Canada (French)
DK	Denmark
FR	France
GR	Germany
IT	Italy
LA	Latin America
NL	Netherlands
NO	Norway
PO	Portugal
SF	Switzerland (French)
SG	Switzerland (German)
SP	Spain
SV	Sweden
SU	Finland
UK	Great Britain
US	United States

If you must define a special keyboard for your computer, use the KEYB command from your AUTOEXEC.BAT file. It's also possible to install a new keyboard with the CONFIG.SYS file. Use the following line:

```
INSTALL=C:\DOS\KEYB.COM UK
```



AUTOEXEC.BAT file contains commands which execute automatically when the computer is switched on

```
PATH=C:\DOS
C:\DOS\KEYB CF,437, C:\DOS\KEYBOARD.SYS
SET COMSPEC=C:\DOS\COMMAND.COM
BREAK ON
PROMPT $P$G
SET TEMP=C:\TMP
```

Configures the path

Specifies keyboard driver and country code

Specifies command interpreter

Enables break option for use during transfer operations

Specifies appearance of DOS prompt

Specifies directory for temporary file storage

```
C:\>
```

End of file

You can now access computer

Include keyboard settings in the AUTOEXEC.BAT file

Setting the speed of the keyboard

You can use the MODE command to set both the speed and the delay rate of most keyboards. However, not all computers recognize this command function. These settings are only possible on AT class computers with programmable keyboards.

The following is the command line for setting the values:

```
MODE CON Rate=x Delay=y
```

"Rate" defines the key repetition rate, which can be between 1 and 32. The larger the specified value, the faster the cursor moves when a key is held down. Rate=1 repeats the key two times per second, Rate=32 repeats the key approximately 30 times per second.

"Delay" defines the time that passes until the pressed key begins repeating automatically. The values that can be used for delay are 1, 2, 3 and 4, which represent a delay of .25, .50, .75 and 1 second, respectively.

Example

To set a keyboard for the highest possible speed, use the following command:

```
MODE CON Rate=32 Delay=1
```

The keyboard now has a repetition frequency of 30 characters per second and a delay of 1/4th second. The default values are 20 for IBM AT-compatible keyboards and 21 for IBM PS/2-compatible keyboards. The default delay is 2 or 1/2 second. If either value is changed, both must be set.

10.6 Defining New Disk Drives

The DRIVER.SYS device driver can be used to define new disk drives. DOS usually recognizes existing disk drives automatically and installs them accordingly.

Being able to define new disk drives is important because disk drives can be operated in various ways. For example, you can use DRIVER.SYS to define a "logical" drive that makes a 1.2 Meg disk appear to be a 360K drive.

Unlike earlier versions of DOS, DRIVER.SYS cannot be used for hard drives in DOS 5.0.

Installing a new disk drive

To assign a new drive indentifier to an existing disk drive or use an external disk drive, type the following command line in CONFIG.SYS:

```
DEVICE=DRIVER.SYS /D:Number /F:Format
```

For "Number" specify either the number of the existing disk drive or external disk drive. A value of 0 corresponds to the first disk drive (A:), 1 corresponds to the second disk drive (B:), etc.

"Format" specifies the type of disk drive according to the following table:

0	160K, 180K, 320K, 360K
1	1.2 Meg (AT disk drive)
2	720K (3.5 inch disk drive)
7	1.44 Meg (3.5 inch disk drive)
9	2.88 Meg (3.5 inch disk drive)

The definition isn't activated until the computer is restarted. When booting, DOS displays the message:

```
External driver loaded for disk drive X
```

For X the computer assigns the first unused disk drive identifier. For example, if the highest drive letter previously assigned is C (for the hard drive), DOS assigns the letter D. The new logical disk drive can be used under disk drive identifier D:.

Example

You can define an existing AT disk drive A: (1.2 Meg) as a 360K disk drive. To do this, add the following line in CONFIG.SYS:

```
DEVICE=DRIVER.SYS /D:0 /F:0
```

/D:0 Zero indicates that DOS should use disk drive A: for the "new" disk drive.

/F:0 Zero indicates that this new disk drive can be addressed as a 360K disk drive.

For example, if you type the above command and DOS displays the following message when booting your computer:

```
External driver loaded for disk drive D
```

you can use drive D: like a 360K disk drive. You no longer need to use the /4 switch when you format 360K diskettes.

Don't be surprised if you see the following two identical lines in the CONFIG.SYS of an AT:

```
DEVICE=DRIVER.SYS /D:0 /F:0
DEVICE=DRIVER.SYS /D:0 /F:0
```

Both command lines define a new 360K disk drive (because of /F:0) based on the existing drive A: (defined by /D:0). This computer could have two new disk drives, D: and E:, that automatically format diskettes at 360K.

Installing and defining a disk drive

When DRIVER.SYS is used to define a disk drive, you can not only define the capacity of the drive (within limitations), but also specify additional parameters, such as the number of tracks and sectors.

To use an existing disk drive under a new label with different settings, add the following line to CONFIG.SYS:

```
DEVICE=DRIVER.SYS /D:Number Parameter
```

The following switches can be used (in combinations) if they are supported by the disk drive:

/C This switch checks whether the diskette lever is open or closed. This enables DOS to determine when you change diskettes. (Use this with AT disk drives.)

/F:Format

Define the type of device here. Instead of "Format", enter the values from the table above.

/H:Heads

Define the number of read or write/heads supported by the disk drive here. You can set from 1 to 99 heads; the default setting is 2.

/S:Sectors

Define the number of sectors per track here. This number depends on the disk drive. You can set from 1 to 99 sectors; the default value is 9.

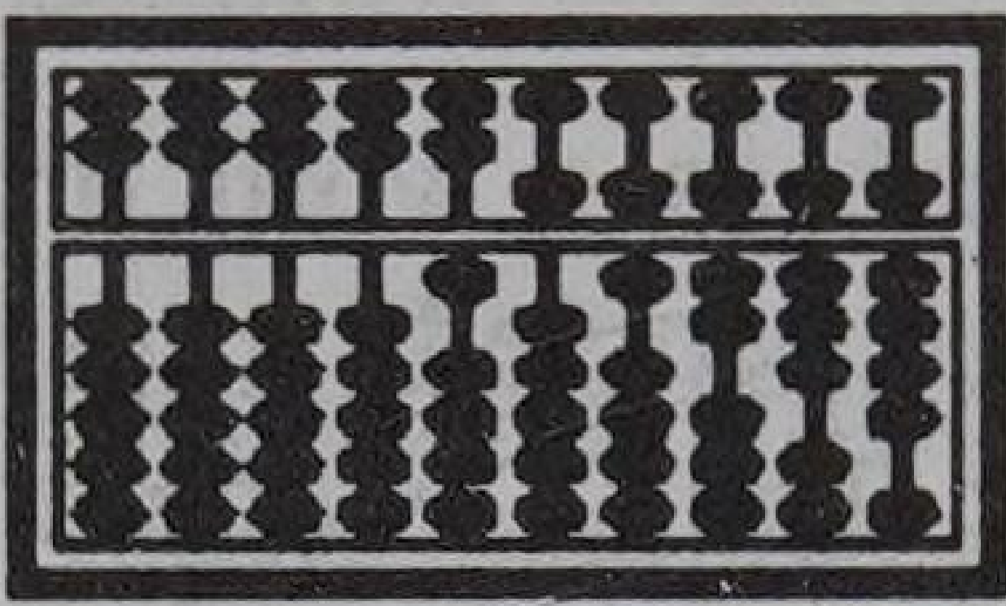
/T:Tracks

Define the number of tracks per page here. This number depends on the disk drive. You can set values from 1 to 999; the default value is 80.

The following table lists possible combinations for the switches:

Disk drive	/F	/T	/S	/H	Comment
360K	0	40	9	2	from PC/XT
720K	2	80	9	2	from PC/XT
1.2 Meg	1	80	15	2	from AT
1.44 Meg	7	80	18	2	from AT
2.88 Meg	7	80	36	2	special disk drive

Chapter



11

Memory Management

Just a short time ago, the amount of memory in a PC wasn't an important consideration. Most computers came with 512K, which was adequate for most users. But even if you had a memory-hungry application, it was still easy to upgrade your system to 640K.

Today, however, it's not unusual for a computer to have more memory. A PC containing from one megabyte to four megabytes of memory is not uncommon.

In this chapter, we'll explain the ins and outs of memory. We'll discuss memory management in detail, so that you can understand and optimize your memory usage.

11.1 Background

Memory is required in order for a computer to work. There are two types of computer memory:

Random Access Memory	(RAM) - Your computer can write and then read data from this type of memory. When computer programs refer to memory, they are almost always talking about this type.
Read-Only Memory	(ROM) - Your computer cannot write to ROM, hence the term read-only memory. This type of memory contains programs and data that are permanent and cannot be changed. ROMs are most often used to instruct hardware devices, such as video graphics adapters, how to operate in your PC.

Both types of memory are used to hold programs and data that the computer processes. The difference is that RAM is volatile - if you switch off your computer, the programs and data in memory are lost, whereas ROM is permanent - if you switch off your computer, the programs and data are retained.

Let's look at RAM more closely. Basically, this is how a program uses this type of memory:

When a program such as a word processor is started, the file containing the program is loaded (i.e., copied) from the diskette or hard drive into the memory area of the computer. The DOS command interpreter tells the computer where in memory to load the program file.

As you type characters using the word processor, the data (characters) is also stored in another area of the computer's memory. The word processor tells the computer where in memory to store this data. The data remains in memory until you tell the word processor to save (i.e., write) the text to a storage device.

When you change the text, you're actually changing the data in the computer's memory. If you switch off your computer without saving the text to a storage device, the text is gone forever.

How much memory does my computer have?

By using the DOS 5.0 MEM command, you can easily determine how much memory your computer has. Type:

```
MEM
```

You'll see a display similar to the following:

```
655360 bytes total conventional memory
655360 bytes available to MS-DOS
542288 largest executable program size
```

The first line tells us that DOS recognizes a total of 640K of memory. Since 1K = 1024 bytes of memory, 640K = 640 * 1024 bytes = 655,360 bytes.

The last line tells us that, in this computer, part of the computer's memory is used for other purposes. So approximately 530K of memory remains to run programs.

If your computer has more than 640K of memory, MEM will display a few additional lines of information perhaps like this:

```
  55360 bytes total conventional memory
 655360 bytes available to MS-DOS
 583472 largest executable program size

1212416 bytes total EMS memory
 819200 bytes free EMS memory

3538944 bytes total contiguous extended memory
      0 bytes available contiguous extended memory
 458732 bytes available XMS memory
        MS-DOS resident in High Memory area
```

The terms EMS, XMS, extended and high memory appear in these lines. We'll discuss these terms in more detail shortly.

11.2 Conventional And Extended Memory

Before discussing conventional and extended memory, let's take a few moments and talk about the early days of PCs.

Early IBM/PC computers were powered by the Intel 8088 and 8086 CPUs. They were capable of referencing or addressing up to 1 Meg of memory. This 1 Meg limit was determined by the number of *address lines* on the 8088 chip. Since the 8088 has 20 address lines, it was able to address $2^{20} = 1{,}048{,}576$ memory (1 Meg) locations.

The 80286 CPU appeared a few years later in the AT model computers. The 80286 has 24 address lines so it is able to address $2^{24} = 16{,}777{,}266$ memory (16 Meg) locations.

The 80386 and 80486 CPUs, which are found in the latest computers, both have 32 address lines. So they are able to address $2^{32} = 4{,}294{,}967{,}296$ (4 gigabytes) locations.

From this historical perspective, we can see that all of the CPUs are able to address at least one megabyte of memory. The original DOS was designed around the 1 Meg limit and this range of memory has become known as *conventional memory*. To distinguish between this first megabyte of memory, additional memory beyond the 1 Meg limit is referred to as *extended memory*.

Since the 8088 and 8086 CPUs don't have enough address lines, computers with these processors cannot use memory beyond the 1 Meg limit. But PCs with 80286, 80386 and 80486 CPUs can use extended memory.

Why 1 Meg adds up to only 640K

If your computer has 1 Meg of memory, a program still may not be able to use this full amount of memory. In fact, most programs can only use up to 640K of memory.

To explain why this occurs, let's use an example. Suppose that you want to construct a small building and you need 1,024 square feet of property. You purchase 1,024 square feet, but later find out that it's actually divided by a six lane freeway that has the right of way.

The freeway divides your property into two sections: one 640 square feet in size and the other 384 square feet. Although you have 1,024 square feet of property, you won't be able to use the 384 square foot section to construct the building according to plan.

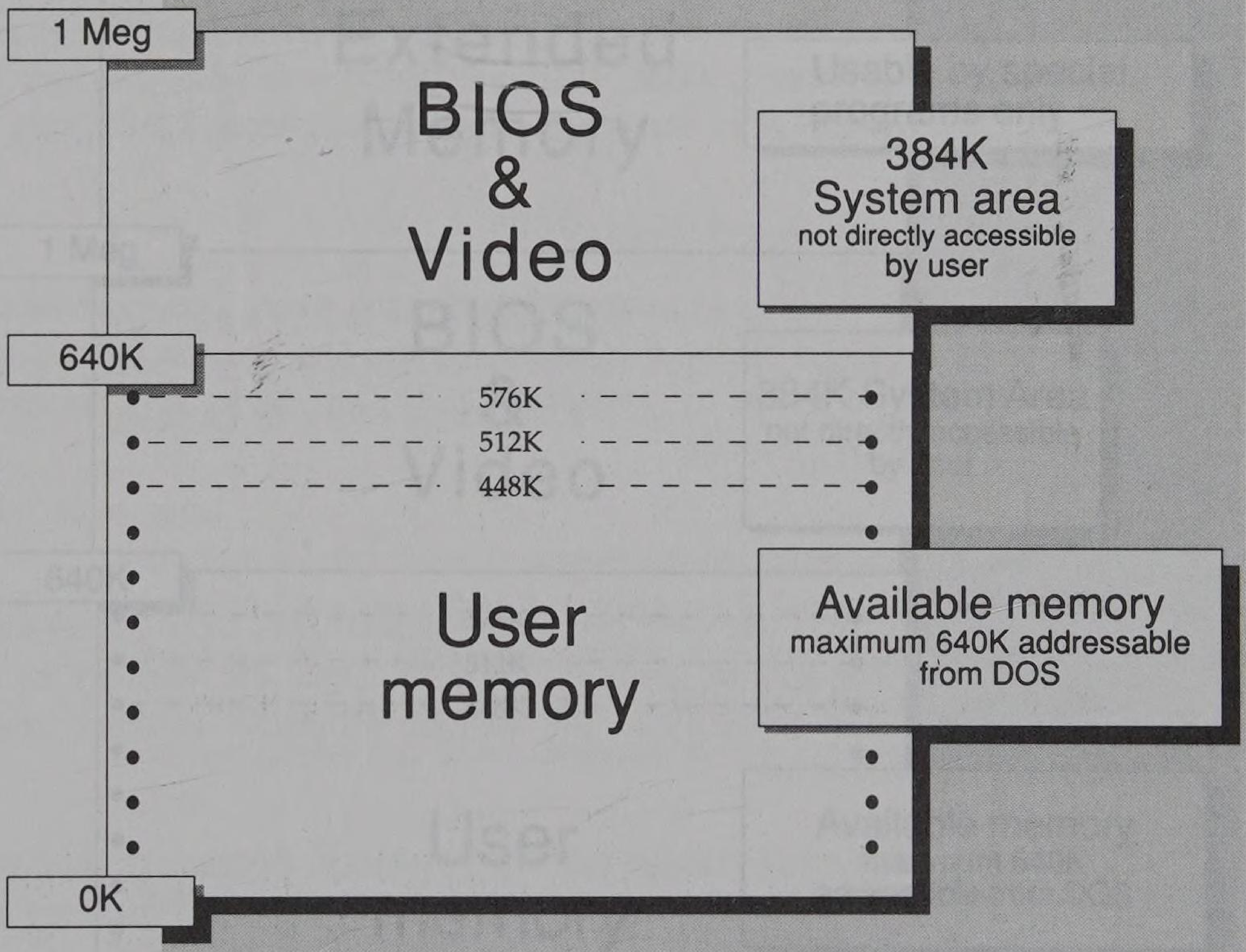
This is similar to how memory appears to DOS. When you load a program, DOS makes memory available to the program in a contiguous (undivided) section.

The maximum amount of memory that DOS can make available to a program is 640K. The other 384K is reserved by DOS. Although your computer may have 1024K of memory, you can use only 640K of it effectively.

More details about conventional memory

Before discussing extended memory further, let's look at conventional memory in more detail.

The figure below shows the layout of conventional memory.


1 Meg
BIOS & Video
384K System area not directly accessible by user
640K
576K
512K
448K
User memory
Available memory maximum 640K addressable from DOS
0K

Conventional memory layout

You'll see that *low memory* ranges up to the 640K limit. This area of memory is sometimes referred to as user memory. Some portions of DOS are stored within low memory.

The remainder of low memory is for user programs and data. If your computer has less than 640K, the amount of user memory for the programs and data is reduced proportionally.

Above the 640K limit, conventional memory ranges up to the 1 Meg limit. The additional 384K of memory is reserved by DOS and is sometimes referred to as *upper memory*.

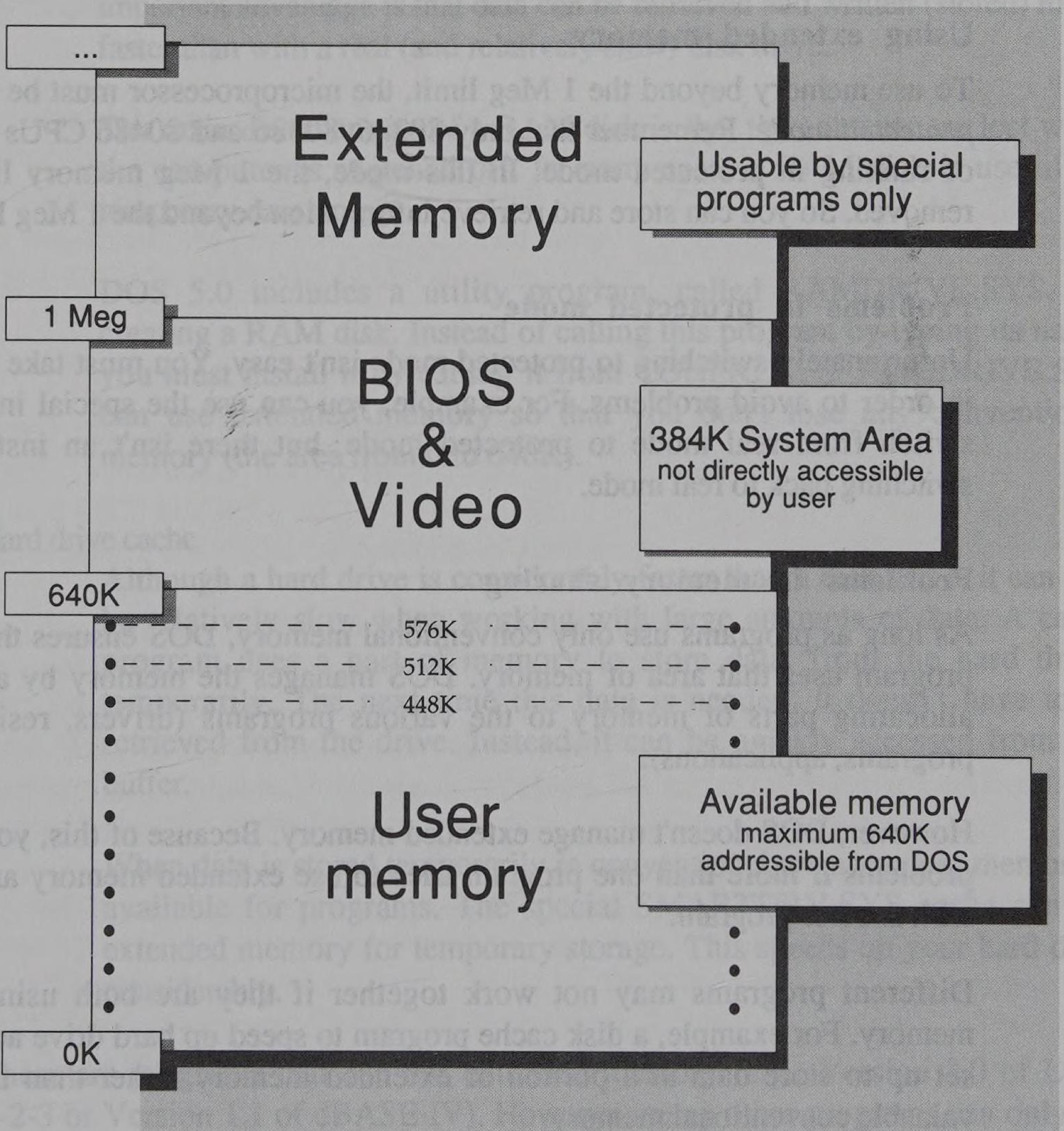
Within upper memory are the BIOS (basic input/output system) ROM, video memory and ROM and other hardware dependent programs (e.g., hard drive controllers, network cards, modem).

Usually only parts of upper memory are used by the computer. This leaves unused sections (or holes) of memory in this address range.

11.3 Extended Memory

A 80286, 80386 or 80486 computer can address much more memory than a 8086 or 8088 computer. As we previously mentioned, the memory above the 1 Meg conventional limit is called extended memory.

In a 80286, up to 16 Meg of memory is addressable. Up to 4 gigabytes of memory is addressable for 80386 and 80486 computers. Unfortunately, DOS can't automatically use this extended memory.


...
Extended Memory
Usable by special programs only
1 Meg
BIOS & Video
384K System Area not directly accessible by user
640K
576K
512K
448K
User memory
Available memory maximum 640K addressible from DOS
0K

Extended memory layout

This is a layout of a typical AT-class computer (computers with 80286, 80386 and 80486 CPUs) with 1 Meg of memory. The layout of conventional memory is identical to that for 8086 and 8088 computers.

The AT has 640K of conventional memory and an area reserved for the BIOS, video and other hardware-dependent devices. The remaining 384K of memory is found in an area above the 1 Meg limit. So this 384K is part of extended memory.

In order to be compatible with the 8086 and 8088 computers, the AT-class computers operate in what is called *real mode*. In this mode, the computer operates as if it were an 8086 computer and therefore can address a maximum of 1 Meg of memory. So an AT-class computer in real mode cannot use memory beyond the 1 Meg limit.

Using extended memory

To use memory beyond the 1 Meg limit, the microprocessor must be switched to *protected mode*. Remember that only 80286, 80386 and 80486 CPUs are capable of running in protected mode. In this mode, the 1 Meg memory limitation is removed. So you can store and retrieve information beyond the 1 Meg line.

Problems in protected mode

Unfortunately switching to protected mode isn't easy. You must take precautions in order to avoid problems. For example, you can use the special instruction to switch from real mode to protected mode, but there isn't an instruction for switching back to real mode.

Problems in memory sharing

As long as programs use only conventional memory, DOS ensures that only one program uses that area of memory. DOS manages the memory by assigning or allocating parts of memory to the various programs (drivers, resident utility programs, applications).

However, DOS doesn't manage extended memory. Because of this, you may have problems if more than one program tries to use extended memory and conflicts with another program.

Different programs may not work together if they are both using extended memory. For example, a disk cache program to speed up hard drive access can be set up to store data in a portion of extended memory rather than taking away valuable conventional memory.

However, if your word processor is capable of using extended memory, it may try to store text into the same area. Both the word processor and disk cache program may overwrite each other's data and eventually cause the computer to crash.

There are a few ways to avoid these kinds of conflicts. For example, a disk cache program might use a portion of extended memory and then fool other programs into thinking that this memory area doesn't exist. Unfortunately, since extended memory isn't managed systematically the way DOS manages conventional memory, the problem may still occur.

Which DOS programs use extended memory

Two DOS 5.0 utilities can use extended memory:

RAM disk

A RAM disk is a pseudo disk drive that uses a part of the memory for storing data. There are several advantages to using a RAM disk. The most important advantage is that data can be retrieved and written (stored) much faster than with a real (and relatively slow) disk drive.

The main disadvantage of a RAM disk is that the contents are lost when the computer is switched off. In many cases a RAM disk is useful for temporary data storage.

DOS 5.0 includes a utility program, called RAMDRIVE.SYS, for creating a RAM disk. Instead of calling this program by typing its name, you must install it by calling it from CONFIG.SYS. RAMDRIVE.SYS can use extended memory so that you don't lose any conventional memory (the area from 0 to 640K).

Hard drive cache

Although a hard drive is considerably faster than a disk drive, it can still be relatively slow when working with large amounts of data. A cache program uses a part of memory to store data, from the hard drive, temporarily. The next time this data is needed, it doesn't have to be retrieved from the drive. Instead, it can be quickly accessed from this buffer.

When data is stored temporarily in conventional memory, less memory is available for programs. The special SMARTDRV.SYS cache can use extended memory for temporary storage. This speeds up your hard drive considerably.

There are programs that can use extended memory (e.g., Version 3.0 of Lotus 1-2-3 or Version 1.1 of dBASE IV). However, many other programs can only use only extended memory as expanded memory.

11.4 Expanded Memory

Extended memory has two disadvantages:

1. It can only be used on AT-class computers. Therefore millions of 8086 and 8088 computer users cannot take advantage of any memory beyond the 1 Meg limit.

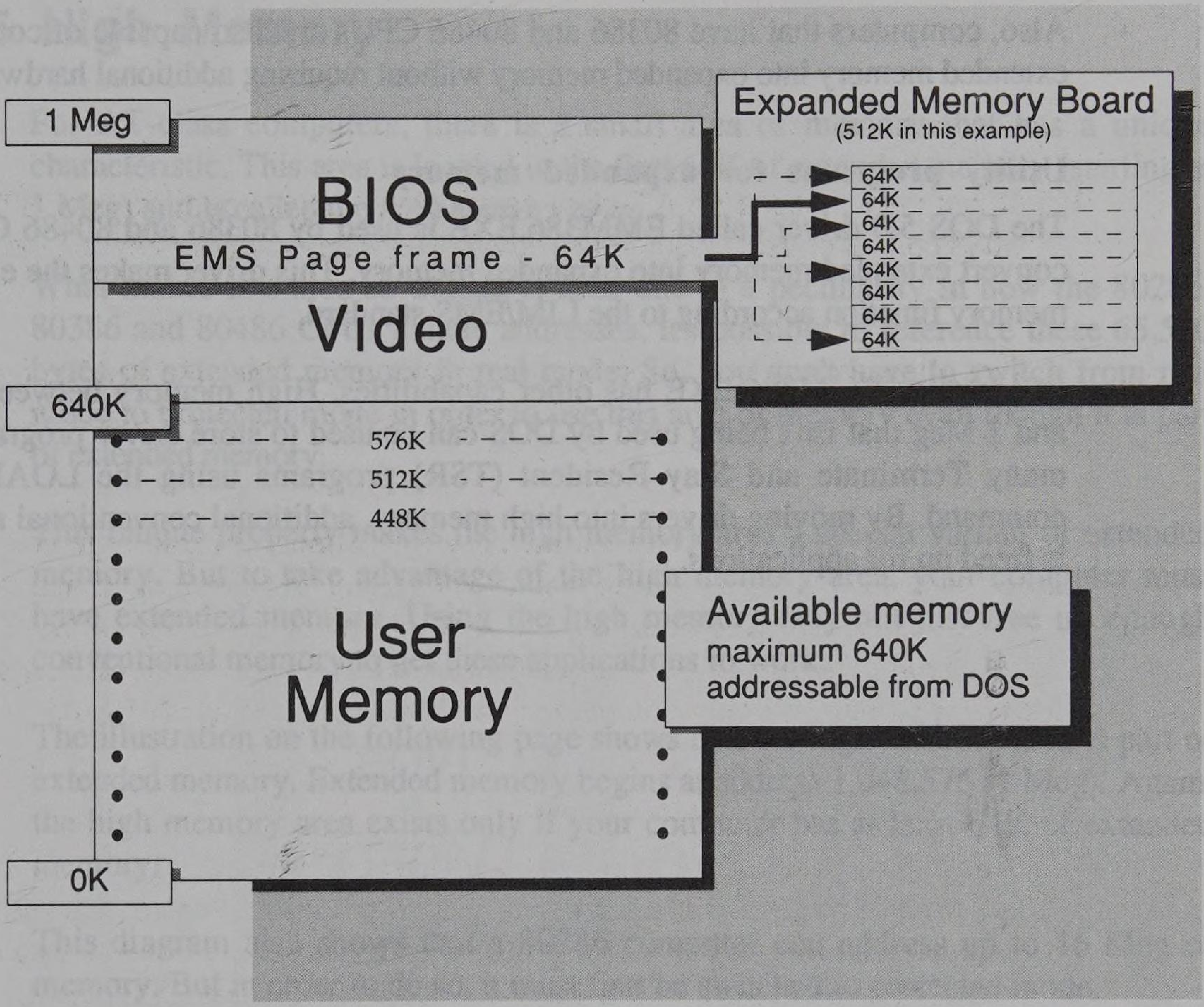
2. Not all programs have been written to take advantage of extended memory. Therefore only a small universe of programs are able to address beyond the 1 Meg limit.

To try to solve these two extended memory limitations, several large companies joined forces to develop a uniform method for adding and using additional memory on the PC. This method is called the Expanded Memory Specification (EMS). The initials of the participating companies (Lotus, Intel and Microsoft) are sometimes added to the name, so you might see the standard referred to as LIM/EMS.

How does expanded memory work?

Expanded memory is a combination of hardware and software. The memory itself is contained on a printed circuit board with special electronics. One principle behind expanded memory is that it can be used by 8086 and 8088 computers. So memory that uses this scheme must be addressable below the 1 Meg limit.

An area in upper memory, between 640K and 1 Meg, is designated as a *page frame*. The page frame is a 64K area of memory that acts as a window to a much larger area of expanded memory. The electronic circuitry lets you select the part of expanded memory that appears in this window at any given time. Just as DOS manages memory up to the 640K limit, special software called an EMS driver manages the expanded memory. When a program references a section of expanded memory that isn't in the window, the EMS software/hardware combination replaces an unneeded section of memory out of the page frame with the section that is needed.


Expanded Memory Board
(512K in this example)
64K
64K
64K
64K
64K
64K
64K
64K
1 Meg
BIOS
EMS Page frame - 64K
Video
640K
576K
512K
448K
User
Memory
Available memory
maximum 640K
addressable from DOS
0K

Expanded memory layout

The above figure shows the layout of expanded memory. The expanded memory on the EMS circuit board is completely independent of the computer's conventional memory (and extended memory, if present). The electronic circuitry on the board manages the data in the EMS memory.

Of course, a program must be especially written to be able to use expanded memory. But many programs are already set up to recognize and take advantage of a computer's expanded memory. Some of the major applications that can use expanded memory are: dBASE IV, Framework, Lotus 1-2-3, Microsoft Windows, Microsoft Word, Microsoft Works, Turbo Pascal and WordPerfect. There are many other applications that can use expanded memory.

Expanded memory - some variations

Earlier we said that you needed an EMS printed circuit board to use expanded memory. Computers that contain the NEAT chip set are already capable of using expanded memory without additional hardware. By adding a special EMS software driver, a computer's extended memory will behave as if it were expanded memory.

Also, computers that have 80386 and 80486 CPUs are also capable of converting extended memory into expanded memory without requiring additional hardware.

Utility programs for expanded memory

The DOS 5.0 driver called EMM386.EXE is used by 80386 and 80486 CPUs to convert extended memory into expanded memory. This driver makes the extended memory function according to the LIM/EMS standard.

In addition, EMM386.EXE has other capabilities. High memory between 640K and 1 Meg that isn't being used by DOS can be used to store driver programs and many Terminate and Stay Resident (TSR) programs using the LOADHIGH command. By moving drivers into high memory, additional conventional memory is freed up for applications.

11.5 High Memory

For AT-class computers, there is a small area of memory that has a unique characteristic. This area is located in the first 64K of extended memory (starting at 1 Meg) and is called the *high memory area.*

What makes this area so unique is that, due to a peculiarity in how the 80286, 80386 and 80486 CPUs handle addresses, it's possible to reference these 65,520 bytes of extended memory in real mode. So, you don't have to switch from real mode to protected mode in order to use this area of memory even though it is part of extended memory.

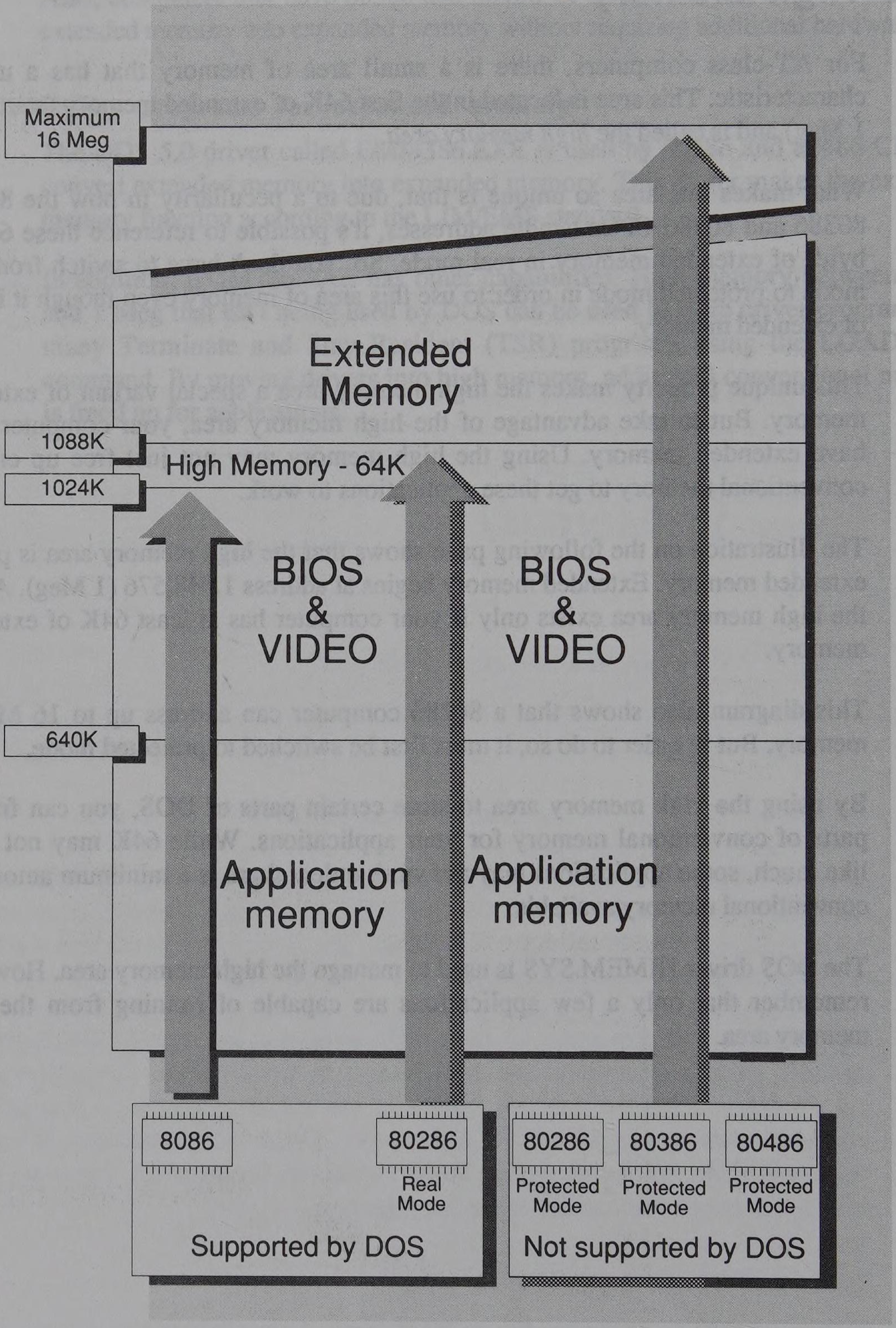
This unique property makes the high memory area a special variant of extended memory. But to take advantage of the high memory area, your computer must have extended memory. Using the high memory may not just free up enough conventional memory to get these applications to work.

The illustration on the following page shows that the high memory area is part of extended memory. Extended memory begins at address 1,048,576 (1 Meg). Again, the high memory area exists only if your computer has at least 64K of extended memory.

This diagram also shows that a 80286 computer can address up to 16 Meg of memory. But in order to do so, it must first be switched to protected mode.

By using the high memory area to store certain parts of DOS, you can free up parts of conventional memory for your applications. While 64K may not seem like much, some applications may not work unless there is a minimum amount of conventional memory available.

The DOS driver HIMEM.SYS is used to manage the high memory area. However, remember that only a few applications are capable of running from the high memory area.

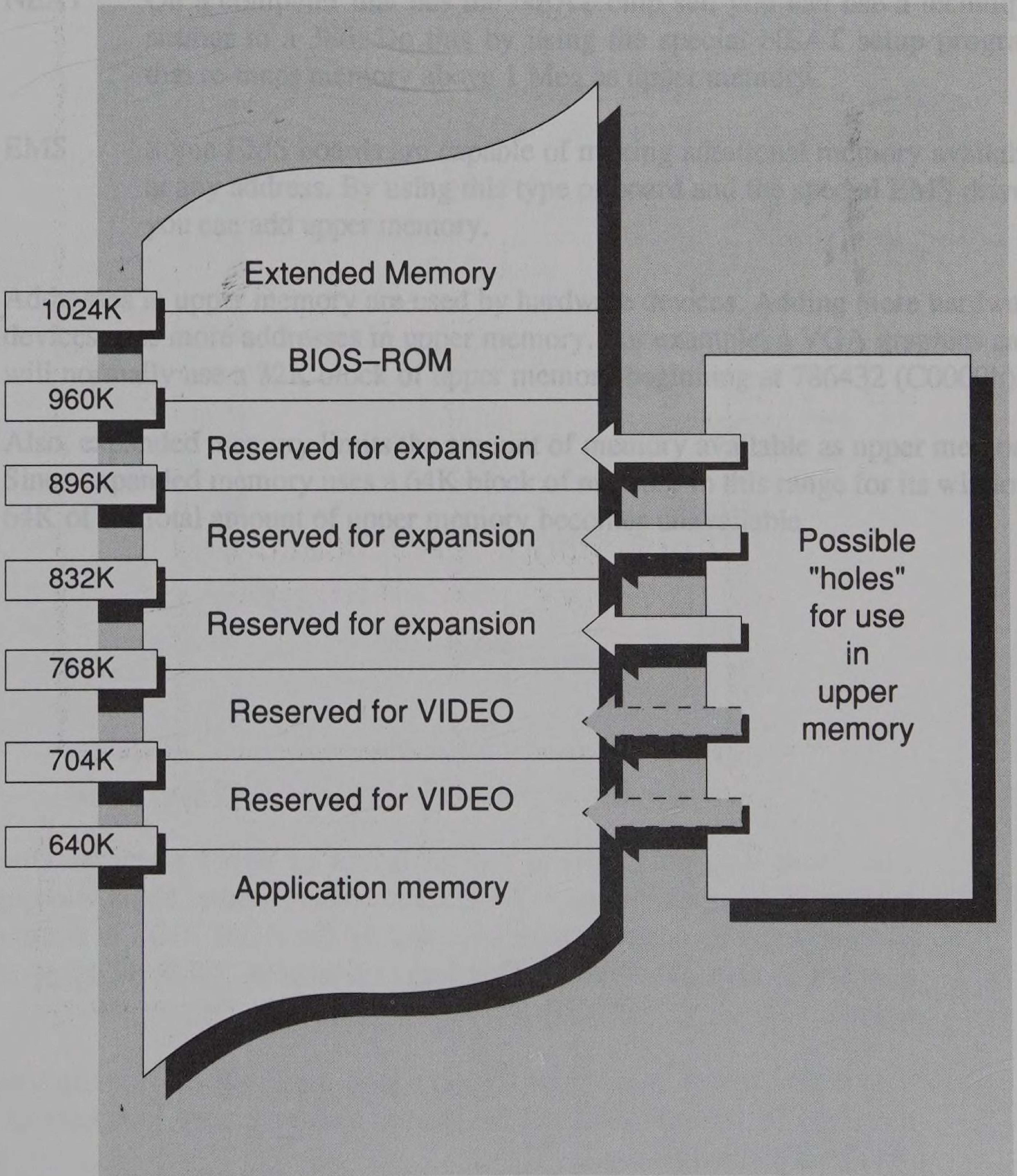

Maximum 16 Meg
Extended Memory
1088K
High Memory - 64K
1024K
BIOS & VIDEO
BIOS & VIDEO
640K
Application memory
Application memory
8086
80286
Real Mode
80286
Protected Mode
80386
Protected Mode
80486
Protected Mode
Supported by DOS
Not supported by DOS

Special features of high memory

11.6 Upper Memory

The memory above the 640K line and below the 1 Meg is referred to as the upper memory blocks or *upper memory*. These areas are most often used for hardware-related memory requirements. For example, video graphics cards, network adapters and specialized hard drive controllers use memory addresses in upper memory to control their devices.

But upper memory is never completely filled. There are always "holes" in this range that are unused. On most computers, you can fill these "empty" areas by adding RAM chips. And since this range is below the 1 Meg line, 8086 and 8088 computers can address this area.


Extended Memory
1024K
BIOS–ROM
960K
Reserved for expansion
896K
Reserved for expansion
832K
Reserved for expansion
768K
Reserved for VIDEO
704K
Reserved for VIDEO
640K
Application memory
Possible "holes" for use in upper memory

Special features of upper memory

The previous diagram shows the layout of memory between 640K and 1024K (1 Meg). On most computers, this area of memory is reserved for hardware add-ons. But if an area isn't being used, then the "hole" can be used.

With DOS 5.0 you can, under certain conditions, use many areas of upper memory just as you would use memory below the 640K limit. The LOADHIGH command is used to load a program into upper memory and the DEVICEHIGH command is used to load drivers into upper memory.

11.7 Requirements For Using Upper Memory

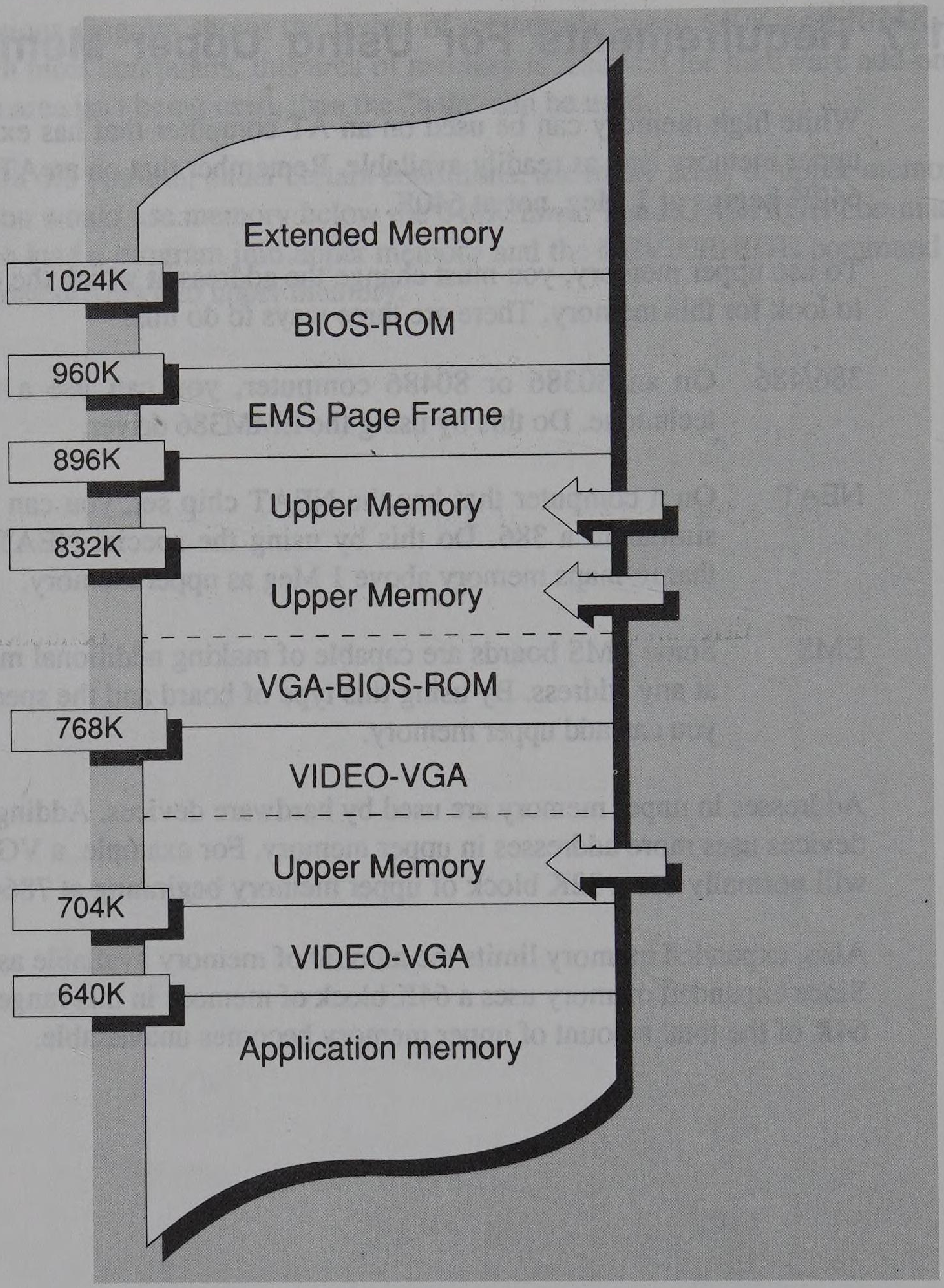
While high memory can be used on an AT computer that has extended memory, upper memory isn't as readily available. Remember that on an AT, memory above 640K begins at 1 Meg, not at 640K.

To use upper memory, you must change the address at which the computer begins to look for this memory. There are three ways to do this:

386/486	On an 80386 or 80486 computer, you can use a virtual memory technique. Do this by using the EMM386 driver.
NEAT	On a computer that has the NEAT chip set, you can use a technique similar to a 386. Do this by using the special NEAT setup program that re-maps memory above 1 Meg as upper memory.
EMS	Some EMS boards are capable of making additional memory available at any address. By using this type of board and the special EMS driver, you can add upper memory.

Addresses in upper memory are used by hardware devices. Adding more hardware devices uses more addresses in upper memory. For example, a VGA graphics card will normally use a 32K block of upper memory beginning at 786432 (C0000h).

Also, expanded memory limits the amount of memory available as upper memory. Since expanded memory uses a 64K block of memory in this range for its window, 64K of the total amount of upper memory becomes unavailable.


Extended Memory
1024K
BIOS-ROM
960K
EMS Page Frame
896K
Upper Memory
832K
Upper Memory
VGA-BIOS-ROM
768K
VIDEO-VGA
Upper Memory
704K
VIDEO-VGA
640K
Application memory

Upper memory with VGA card and EMS page frame

The above diagram shows a typical layout of upper memory. One block of memory is occupied by the VGA graphics card. Another block is occupied by an EMS board. The address range occupied by the ROM BIOS at the end of upper memory is also unavailable. This leaves a total of 128K of upper memory as unused.

You'll also notice that upper memory isn't available as a contiguous block of memory. To take advantage of the unused upper memory, you have to be able to use the disjointed blocks.

11.8 What Is XMS Memory?

Now that we've discussed high memory and upper memory, we can explain another computer memory term: XMS.

XMS, which is an abbreviation for eXtended Memory Specification, is a way to administer the use of extended memory, high memory and upper memory. To simplify matters, we can divide memory administration tasks as follows:

- low or user memory - managed by DOS
- EMS memory - managed by the EMS driver
- upper, high and extended memory - managed by the XMS driver

This means that the XMS driver is an important part of the memory scheme. Prior to XMS, different applications used extended memory independently, which led to program incompatibilities.

DOS doesn't manage this memory directly. Instead, the driver program HIMEM.SYS is used to manage the memory.

11.9 Customizing Memory To Your Requirements

DOS 5.0 is capable of providing more conventional memory up to the 640K line than earlier versions of DOS. This is possible because of the skillful use of memory above 640K - either in upper memory or high memory. As a result, instead of from 520K to 540K of user memory available using DOS 4.01, DOS 5.0 users will find from 590K to 630K available for their applications.

Using high memory

If extended memory - beginning at 1 Meg - is available, DOS can use the first 64K as the high memory area and store part of itself there.

High memory requires extended memory. For example, if you use a memory manager on an 80386 or 80486 to convert all of extended memory to expanded memory, then extended memory no longer exists. To be able to use the high memory area, leave 64K of the extended memory unused by adjusting the EMM386.EXE driver.

In order for DOS to use high memory, the following conditions must be met:

- The computer must have extended memory beginning at 1 Meg. This is only possible on ATs (80286 and upwards).

- A special driver program must make the first 64K of extended memory available as high memory. With DOS 5.0, use the XMS driver HIMEM.SYS for ATs. Some special memory management programs also have this capability.

  In any case, start this driver program as early as possible in the CONFIG.SYS file. HIMEM.SYS must even be started before EMM386.EXE is called. Then you can start the drivers that use EMS or XMS memory.

- A special command line in the CONFIG.SYS file must be used to instruct DOS to use the high memory for storing the DOS kernal:

  ```
  DOS=HIGH
  ```

  Entering the command line DOS=LOW in the CONFIG.SYS file tells DOS that high memory shouldn't be used directly for DOS. This allows Windows 3.0 to use the high memory area.

The location of the line, DOS=HIGH/LOW, in the CONFIG.SYS file is not important.

Example

Suppose that you want DOS 5.0 to use high memory on an AT with a total of 1 Meg of available memory. To do this, type the following lines at the start of the CONFIG.SYS file:

```
DEVICE=HIMEM.SYS
DOS=HIGH
```

The remaining memory from 640K and above is available as XMS memory.

Use the MEM command to check the available memory. MEM also displays whether part of DOS is stored in high memory. The screen output of MEM might look like the following on an AT with extended memory and the DOS kernal in high memory:

```
 655360 bytes total conventional memory
 655360 bytes available to MS-DOS
 597296 largest executable program size

2097152 bytes total contiguous extended memory
 292864 bytes available contiguous extended memory
1427456 bytes available XMS memory
        MS-DOS resident in High Memory Area
```

Using upper memory

To use memory as upper memory, you must start the EMM386.EXE driver in the CONFIG.SYS file. The XMS driver must be loaded ahead of the EMM386 driver. The following is the correct order for using these drivers in the CONFIG.SYS file:

```
DEVICE=HIMEM.SYS
DEVICE=EMM386.EXE
```

As a rule, EMM386 automatically converts all of the existing extended memory to expanded memory. Since this isn't always desirable, the parameter, NOEMS, tells the driver not to allocate any EMS memory but still allocate upper memory.

To use the extended memory as XMS memory and still use EMM386 to provide upper memory, call both driver programs as follows:

```
DEVICE=HIMEM.SYS
DEVICE=EMM386.EXE NOEMS
```

You must also tell DOS to reserve all of the existing upper memory and make it available as "ordinary" conventional memory (like the memory up to 640K).

```
DOS=HIGH,UMB
```

The abbreviation UMB represents the upper memory block. For example, to reserve upper memory and high memory for DOS on a 80386 or higher computer and use 512K for expanded memory, use the following command lines (provided that you have enough memory):

```
DEVICE=HIMEM.SYS
DEVICE=EMM386.EXE 512
DOS=HIGH,UMB
REM Other lines of the CONFIG.SYS follow
```

To use upper memory without storing the DOS kernal (the most important part of DOS, which must always be available in the memory) in high memory, use the line:

```
DOS=LOW,UMB
```

You can use NOUMB instead of UMB to prevent DOS from using the upper memory area as memory. Usually you won't have to specify NOUMB because:

```
DOS=LOW,NOUMB
```

is the default setting if there aren't any parameters specified in CONFIG.SYS.

We recommend using upper memory for applications as long as your programs are able to use it without any problems. If programs have problems using the memory area between 640K and 1 Meg, use NOUMB as your setting.

If necessary, use special memory management programs to "highload" drivers and TSR programs so that you can utilize upper memory.

LOADFIX, which is a special command that is available with DOS 5.0, can be used to load programs that aren't usually run in extra conventional memory.

This command can also be used to load programs that generate a "Packed file corrupt" error message when trying to load into low memory.

LOADFIX can also load some programs that won't load if a network is installed and running.

11.10 Partitioning Memory

As we've seen, DOS 5.0 is very flexible in managing your computer's memory. The following are some examples of the options that are available. Remember that a few of these options are mutually exclusive (can't be used together).

- On a computer with only 640K of memory, neither of these new options (upper memory or high memory) are available. Since DOS 5.0 uses more conventional memory on this kind of computer, less will be available for application programs.

- On an AT class computer or a 386 with 1 Meg of memory, you can either use this memory for shadowing or as extended memory (including high memory).

  If you use 64K of high memory, there is 64K less of extended memory for a RAM disk or a cache program. The computer must have either an 80386 microprocessor, 80486 microprocessor or a NEAT chip set to use upper memory.

- Computers with more than 1 Meg of memory can use both upper memory and extended memory (including high memory). You lose up to 384K of extended memory for upper memory (even if the ROM areas are accelerated by shadowing) and another 64K for high memory. The rest of the memory is available for programs.

The following table shows which options are applicable for different computer equipment:

Computer equipment	High Memory	Extended	Upper Memory
XT/AT up to 640K	-	-	-
AT with 1 Meg	yes	320K as XMS	-
386 with 1 Meg	yes	320K as XMS	-
or			
386 with more than 1 Meg	yes	yes as XMS	yes

Chapter 12

The EDLIN Editor

We've created several text files so far in this book. We've suggested using various editing programs and even introduced you to the EDLIN editor. With the introduction of MS-DOS 5.0, there is a much better full-screen editor that you can use - EDIT. However, if you want to create files using ANSI codes or other special purposes files, sometimes EDLIN is a good alternative.

This chapter demonstrates the basics of using EDLIN, which is a part of every MS-DOS system. EDLIN is a simple *line editor* which lets you create new text files and edit existing ones. Editing a file with EDLIN is much simpler than completely re-entering a file using COPY CON.

To use the examples in this chapter, you'll need the following:

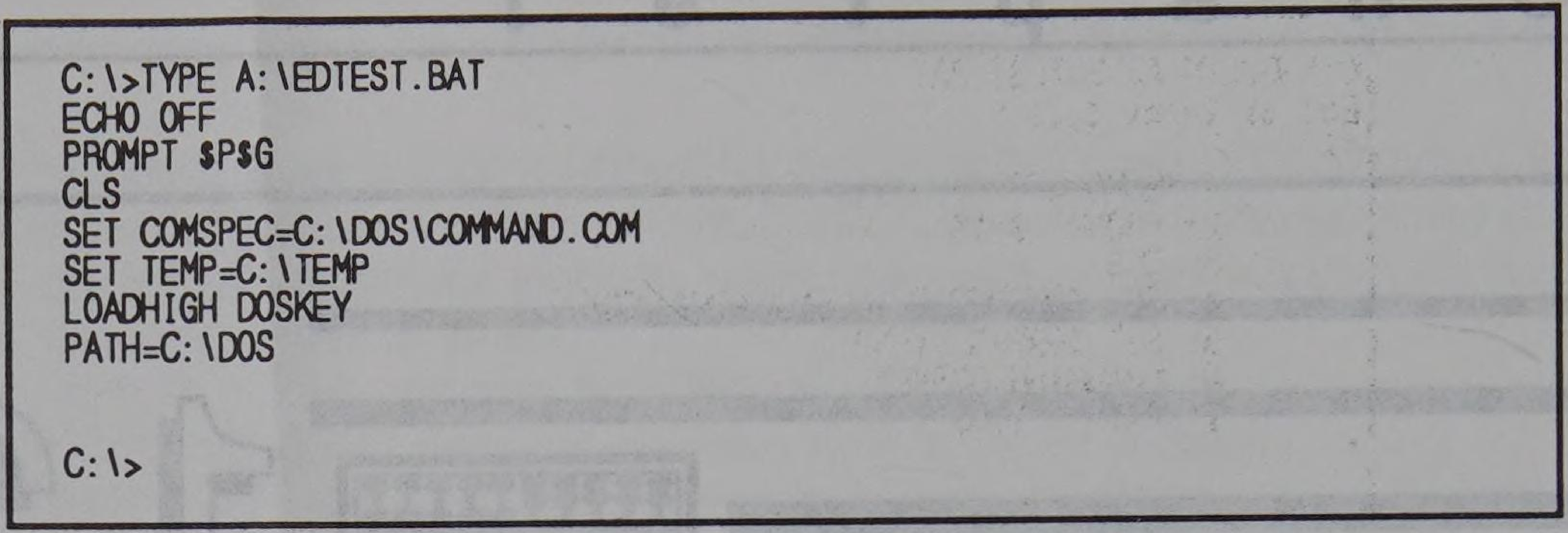
- The companion diskette for this book
- EDLIN (which should be in the DOS directory of your hard drive or on your DOS diskette)

12.1 Editing A Batch File

There's an easier way to create, list and edit files. Make sure the companion diskette is in drive A: and the system prompt currently reads C:>, then type the following:

```
TYPE A:\EDTEST.BAT
```

Then press Enter. The screen displays the EDTEST.BAT file, which will look something like the following:



```
C:\>TYPE A:\EDTEST.BAT
ECHO OFF
PROMPT $P$G
CLS
SET COMSPEC=C:\DOS\COMMAND.COM
SET TEMP=C:\TEMP
LOADHIGH DOSKEY
PATH=C:\DOS


C:\>
```

Listing of EDTEST.BAT using TYPE

Now type the following line:

```
EDLIN A:\EDTEST.BAT
```

Then press Enter. The following appears on the screen:

```
End of input file
*
```

A blinking cursor appears to the right of the asterisk. No system prompt or other data is displayed.

12.1.1 Editing lines with EDLIN

You can't just enter anything in EDLIN. Type the following:

```
What is this?
```

Then press Enter. The PC responds with this or a similar message:

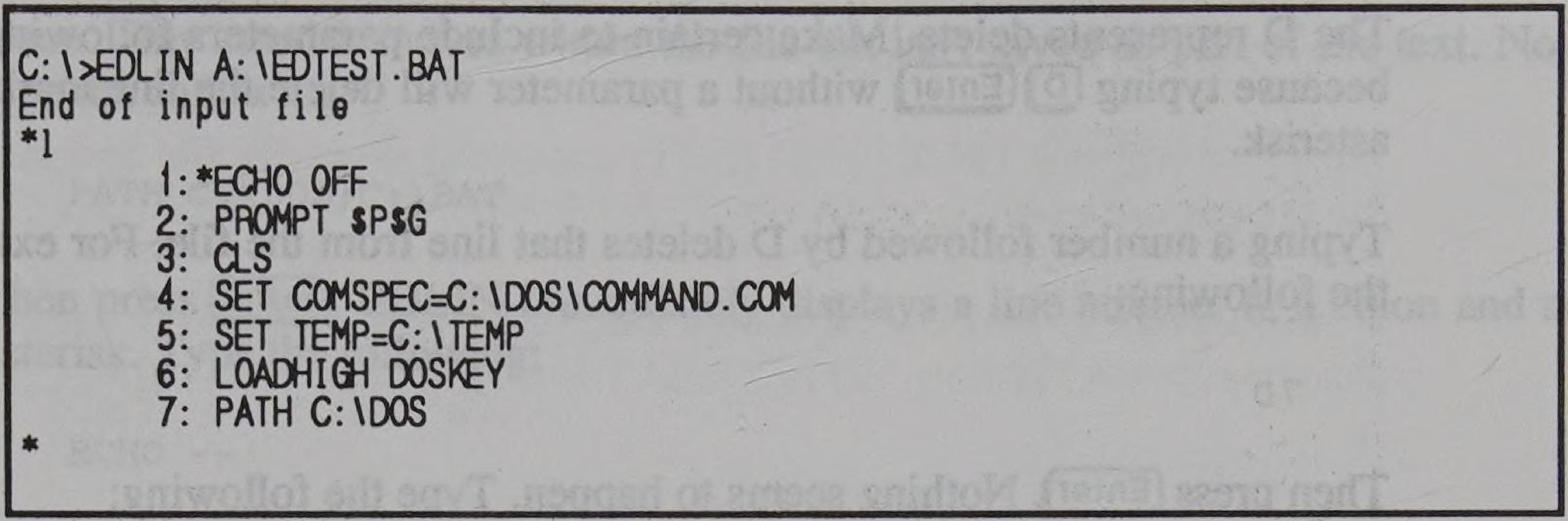
```
Entry error
```

Like MS-DOS, EDLIN only accepts commands that it can understand. For example, EDLIN will display a file if you type the L (List) command.

Type the following:

```
L
```

Then press Enter. Now the lines of text in your EDTEST.BAT file appear numbered on the screen, These lines are indented from the left border of the screen:



```
C:\>EDLIN A:\EDTEST.BAT
End of input file
*l
        1:*ECHO OFF
        2: PROMPT $P$G
        3: CLS
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: SET TEMP=C:\TEMP
        6: LOADHIGH DOSKEY
        7: PATH C:\DOS
*
```

Using the L command to list and number lines in the EDTEST.BAT file

EDLIN displays a file line by line and numbers the lines in the file. The line numbers only appear for your reference in EDLIN.

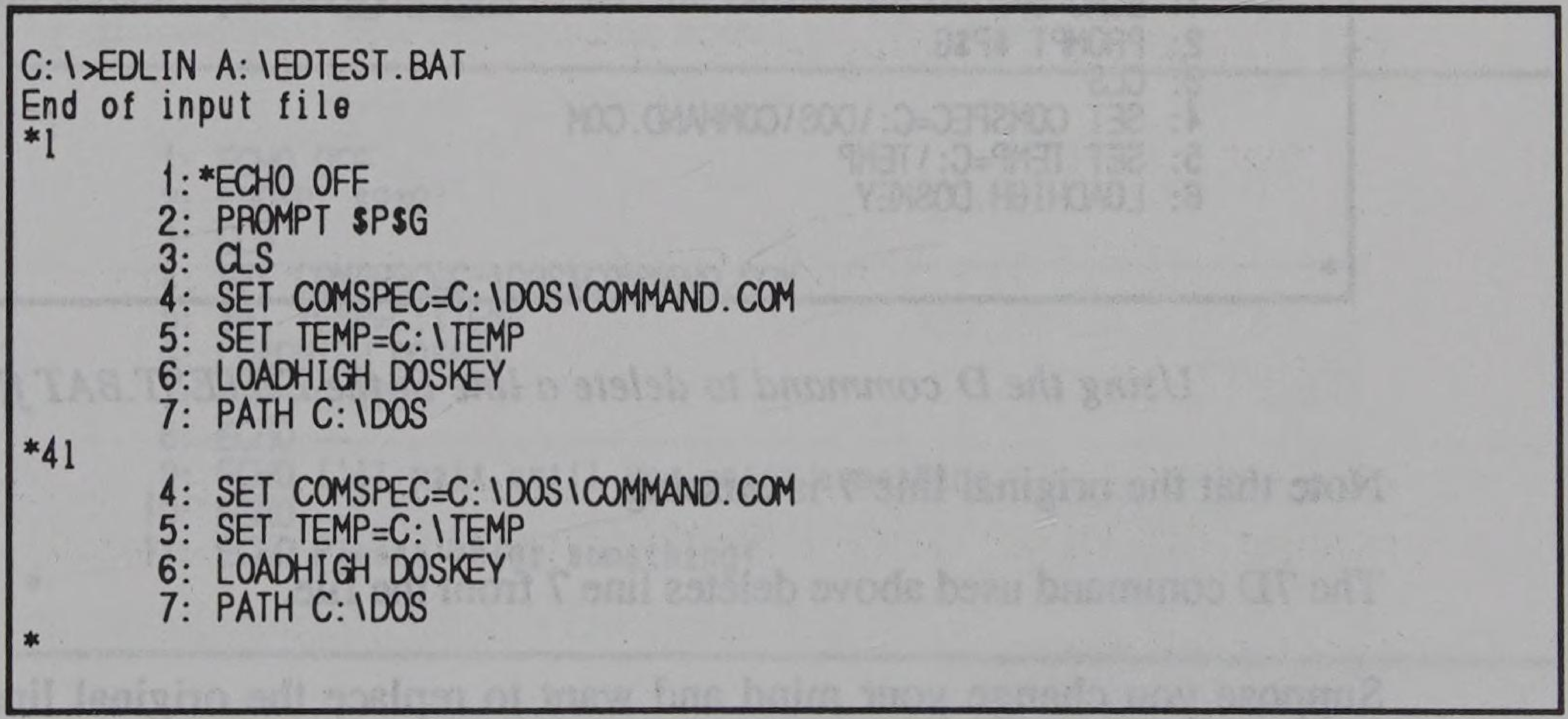
If you created a file in EDLIN, exited EDLIN and used the TYPE command to list the file, these line numbers would not appear on the screen.

Type the following (notice the line number precedes the command letter):

```
4L
```

Then press Enter.

EDLIN lists lines 4, 5, 6 and 7 of the file. Typing L without any other parameters displays the entire file on the screen.



```
C:\>EDLIN A:\EDTEST.BAT
End of input file
*l
        1:*ECHO OFF
        2: PROMPT $P$G
        3: CLS
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: SET TEMP=C:\TEMP
        6: LOADHIGH DOSKEY
        7: PATH C:\DOS
*4l
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: SET TEMP=C:\TEMP
        6: LOADHIGH DOSKEY
        7: PATH C:\DOS
*
```

Using the L command to list specific lines in the EDTEST.BAT file

Suppose you want to change the path for your batch files (line 7). You can delete it using the D command.

The D represents delete. Make certain to include parameters following DELETE because typing D Enter without a parameter will delete the line marked with an asterisk.

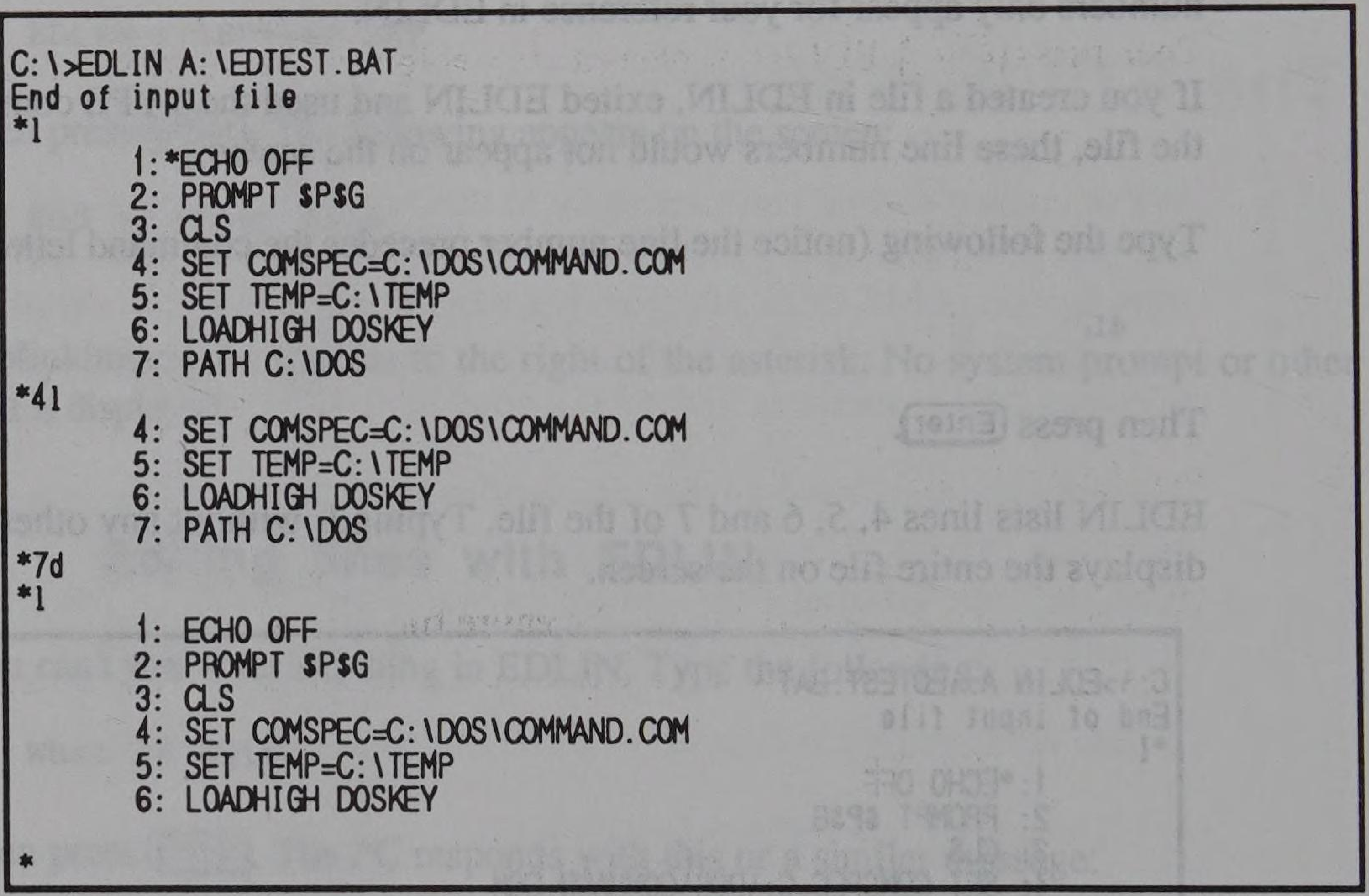
Typing a number followed by D deletes that line from the file. For example, type the following:

```
7D
```

Then press Enter. Nothing seems to happen. Type the following:

```
L
```

Then press Enter. The screen appears as follows:



```
C:\>EDLIN A:\EDTEST.BAT
End of input file
*l
        1:*ECHO OFF
        2: PROMPT $P$G
        3: CLS
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: SET TEMP=C:\TEMP
        6: LOADHIGH DOSKEY
        7: PATH C:\DOS
*4l
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: SET TEMP=C:\TEMP
        6: LOADHIGH DOSKEY
        7: PATH C:\DOS
*7d
*l
        1: ECHO OFF
        2: PROMPT $P$G
        3: CLS
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: SET TEMP=C:\TEMP
        6: LOADHIGH DOSKEY

*
```

Using the D command to delete a line in the EDTEST.BAT file

Note that the original line 7 is missing.

The 7D command used above deletes line 7 from the file.

Suppose you change your mind and want to replace the original line 7 (PATH C:\DOS;C:\BAT) for specifying the path of the batch files. The I (Insert) command lets you insert lines in a text file. Type the following:

```
I
```

Then press Enter. The screen will display a 7, a colon and an asterisk. The asterisk acts as a marker, indicating the line where editing most recently took place. This asterisk moves within the file but isn't saved as part of the text. Now type the following:

```
PATH C:\DOS;C:\BAT
```

Then press Enter. EDLIN immediately displays a line number 8, a colon and an asterisk. Type the following:

```
ECHO --
```

Then press Enter. EDLIN displays a 9, a colon and an asterisk. Type the following:

```
ECHO I'll wait until you enter something
```

Then press Enter. EDLIN displays a 10, a colon and an asterisk. Now type:

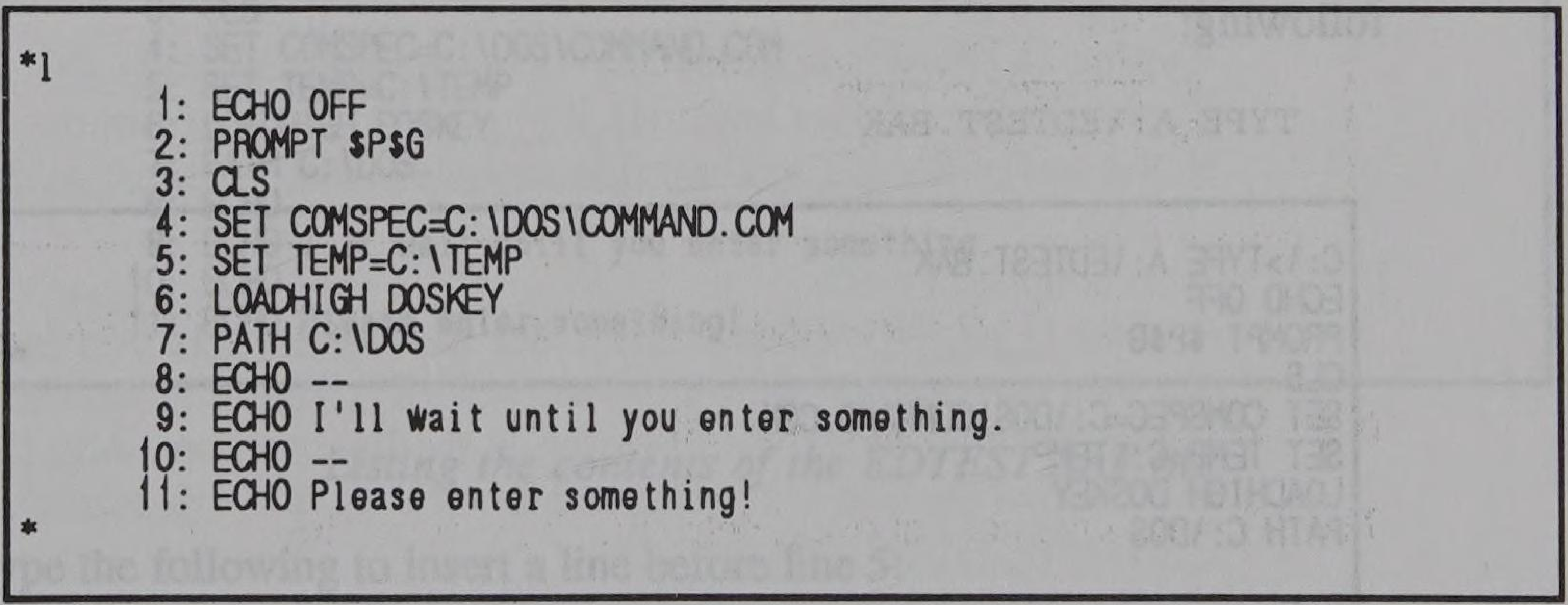
```
ECHO --
```

Then press Enter. EDLIN displays an 11, a colon and an asterisk. Type:

```
ECHO Please enter something!
```

Then press Enter. EDLIN displays a 12, a colon and an asterisk. Press Ctrl + Z or F6, then press the Enter key.

After the cursor jumps to the *command prompt* (the asterisk at the left border of the screen), press L Enter to see the entire file listing:



```
*l
        1: ECHO OFF
        2: PROMPT $P$G
        3: CLS
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: SET TEMP=C:\TEMP
        6: LOADHIGH DOSKEY
        7: PATH C:\DOS
        8: ECHO --
        9: ECHO I'll wait until you enter something.
       10: ECHO --
       11: ECHO Please enter something!
*
```

New lines (8-11) in EDTEST.BAT using the I command

☞ If you forgot to type an ECHO in lines 8 through 11, enter the line number and press Enter. Enter the correct command following the asterisk. Press F6 Enter to return to the command prompt.

This file still isn't saved to disk. So type:

```
E
```

Then press Enter. The E (END EDIT) command saves the file and returns you to the system prompt.

If you wanted to execute this batch file, you would type the following (this will change your PATH since EDTEST.BAT is a copy of an AUTOEXEC.BAT file):

```
A:\EDTEST.BAT
```

Then press Enter. To see the file in the directory, type the following:

```
DIR A:\EDTEST.*
```

Then press Enter. You'll find two files named EDTEST but with two different extensions. As expected, one file has a .BAT extension. However, another file exists under the same name but with a .BAK extension.

EDLIN automatically creates a .BAK (BAcKup) file when a file is edited by EDLIN. Type the following:

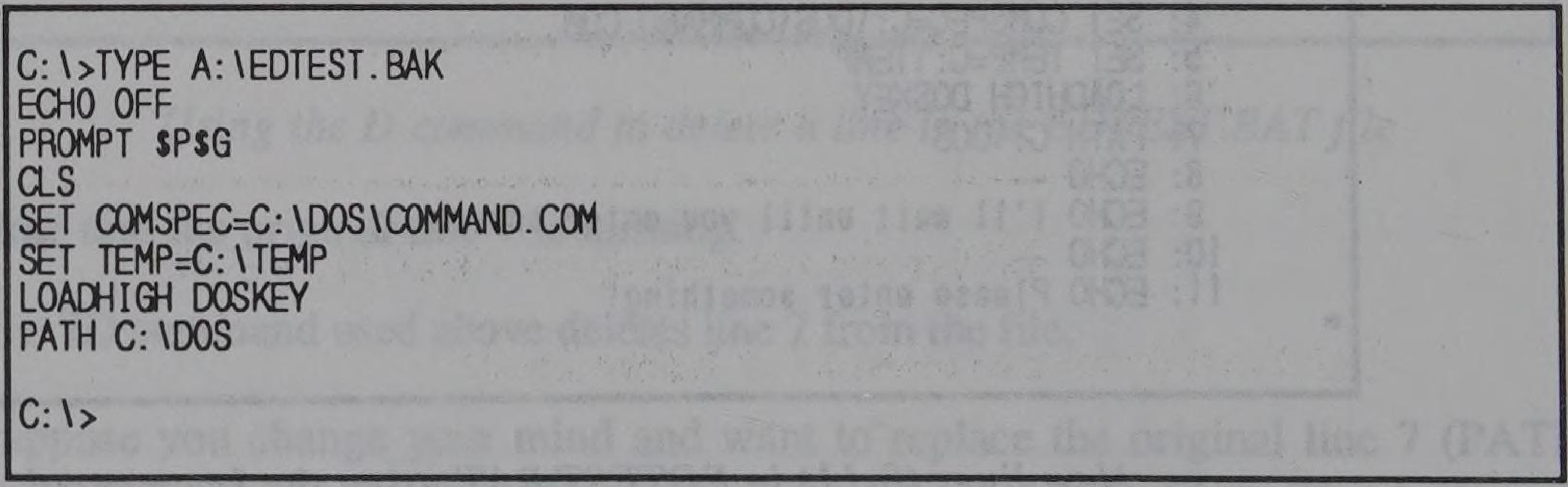
```
EDLIN A:\EDTEST.BAK
```

Then press Enter. The following or a similar error message appears on the screen:

```
Cannot edit .BAK file—rename file
```

Use the TYPE command to check the contents of this backup file. Type the following:

```
TYPE A:\EDTEST.BAK
```



```
C:\>TYPE A:\EDTEST.BAK
ECHO OFF
PROMPT $P$G
CLS
SET COMSPEC=C:\DOS\COMMAND.COM
SET TEMP=C:\TEMP
LOADHIGH DOSKEY
PATH C:\DOS

C:\>
```

Entering TYPE EDTEST.BAK to display the BAK file created by EDLIN

This is the original EDTEST.BAT file before any changes were made. When you change a file with EDLIN and save it again, the previous version is copied to the filename with the .BAK extension.

If for some reason you want to use the original file again, use the RENAME command:

```
RENAME A:\EDTEST.BAK A:\EDTEST.NEW
```

Then you'll be able to edit the batch file again using EDLIN:

```
EDLIN A:\EDTEST.NEW
```

12.1.2 Inserting lines with EDLIN

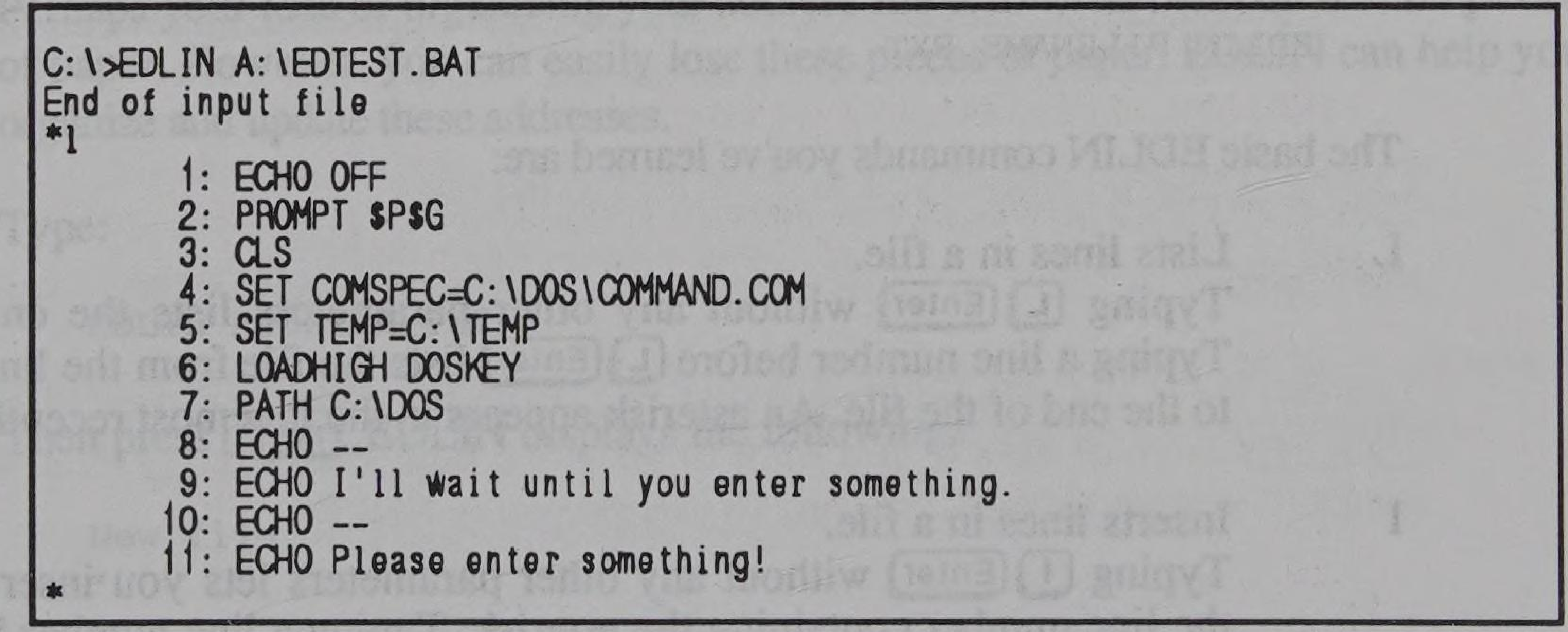
Now we want to make further changes to the EDTEST batch file. Since we saved EDTEST by using the E command, we must use EDLIN again:

```
EDLIN A:\EDTEST.BAT
```

Then press Enter. We learned how to add lines to the end of a file. Now let's add lines to the middle of the file. Type the following to display the file:

```
L
```

Then press Enter. The EDTEST.BAT file appears on the screen:



```
C:\>EDLIN A:\EDTEST.BAT
End of input file
*l
        1: ECHO OFF
        2: PROMPT $P$G
        3: CLS
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: SET TEMP=C:\TEMP
        6: LOADHIGH DOSKEY
        7: PATH C:\DOS
        8: ECHO --
        9: ECHO I'll wait until you enter something.
       10: ECHO --
       11: ECHO Please enter something!
*
```

Listing the contents of the EDTEST.BAT file

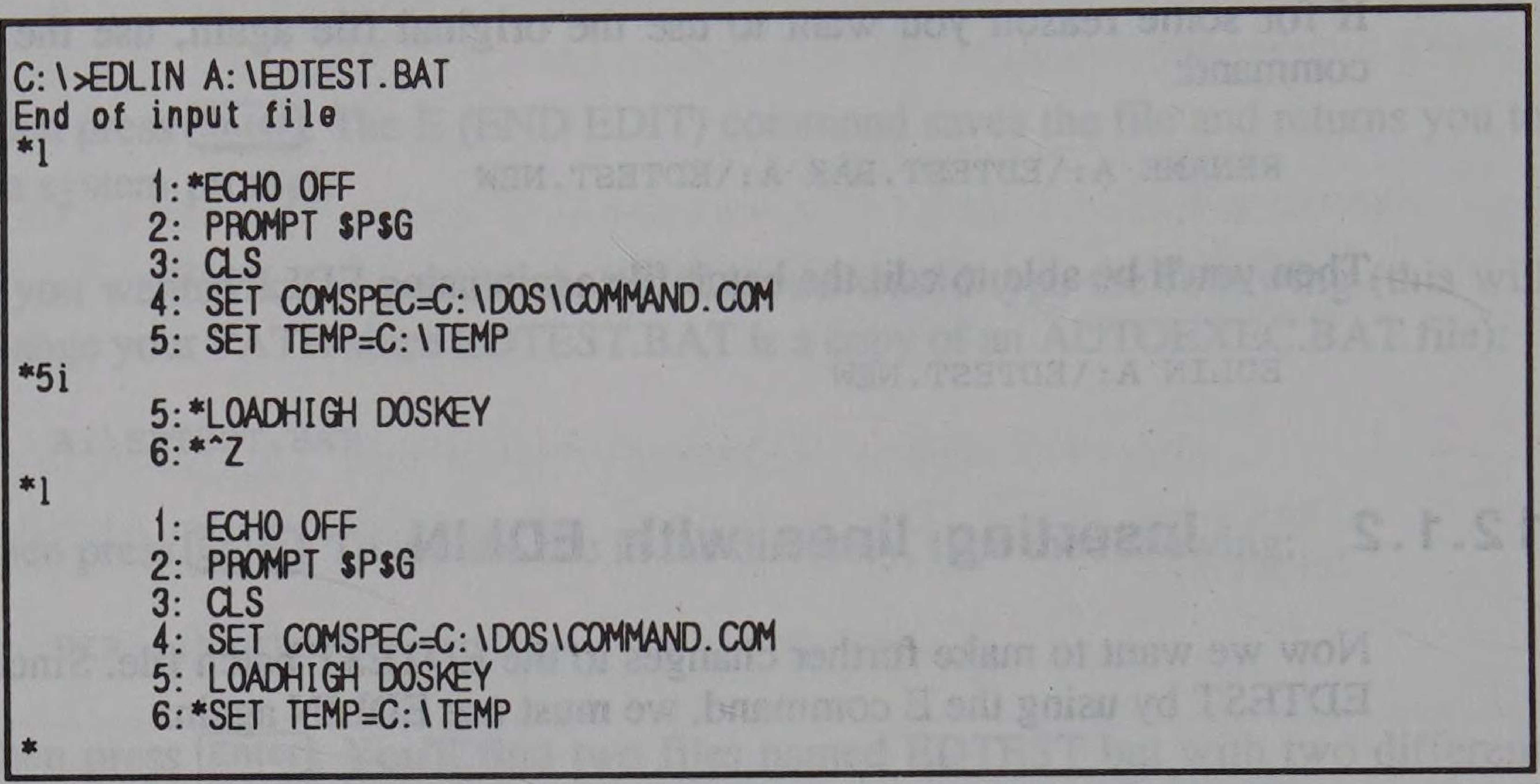
Type the following to insert a line before line 5:

```
5I
```

Then press Enter. EDLIN displays 5:* and the cursor. Type:

```
LOADHIGH DOSKEY Enter
F6 Enter
```

The new line 5 forces the original line 5 and any subsequent lines to move down. Typing L Enter displays the following:



```
C:\>EDLIN A:\EDTEST.BAT
End of input file
*l
        1:*ECHO OFF
        2: PROMPT $P$G
        3: CLS
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: SET TEMP=C:\TEMP
*5i
        5:*LOADHIGH DOSKEY
        6:*^Z
*l
        1: ECHO OFF
        2: PROMPT $P$G
        3: CLS
        4: SET COMSPEC=C:\DOS\COMMAND.COM
        5: LOADHIGH DOSKEY
        6:*SET TEMP=C:\TEMP
*
```

EDTEST.BAT after inserting a new line

Press E Enter to end the edit and return to the system prompt.

Summary

EDLIN is a primitive line editor that enables you to create and edit text files. You can start the editor by typing:

```
EDLIN FILENAME.EXT
```

The basic EDLIN commands you've learned are:

L	Lists lines in a file. Typing L Enter without any other parameters lists the entire file. Typing a line number before L Enter lists the file from the line number to the end of the file. An asterisk appears in the line most recently edited.
I	Inserts lines in a file. Typing I Enter without any other parameters lets you insert lines at the line number containing the asterisk. Typing a line number before I Enter lets you insert lines starting at the line number. Any line numbers higher than the line being inserted are shifted down accordingly. Pressing F6 (or Ctrl C) ends the input.
E	Ends editing and saves the edited file. Also saves the original file with a .BAK extension. This command ends editing, saves the edited file and returns you to the system prompt.
D	Deletes specified lines. Typing n D Enter (where n = a line number) deletes line number n. Typing D Enter without any other parameters deletes the line marked with an asterisk (most recently edited line).

12.2 File Maintenance With EDLIN

EDLIN can also be used in certain cases where you previously used the COPY CON filename. Since EDLIN provides more editing options than COPY CON, you can also use it for general text editing.

12.2.1 Editing files created with COPY CON

You created several text files during our discussion of simple batch files. EDLIN can easily modify these files.

12.2.2 Creating a new file with EDLIN

The following example demonstrates how to use EDLIN as a text editor, which is a program used to create and edit text on the screen.

Suppose that you're getting ready to go on vacation. Since you promised to send postcards to many friends, you'll need a way to organize their addresses.

Perhaps your idea of organizing your address file is to write them on several pieces of paper. However, you can easily lose these pieces of paper. EDLIN can help you organize and update these addresses.

Type:

```
EDLIN ADDRESS.TXT
```

Then press Enter. EDLIN displays the following:

```
New file
*
```

Type:

```
I
```

Then press Enter. The 1:* prompt appears. Press Enter after each line:

```
1: Andrew Bailey, 325 Washington St, Salt Lake City, UT 75680
2: Jill Martin, 788 N. Elm St, San Diego, CA 94008
3: Bill Remington, 104 Ocean View, Santa Barbara, CA 94007
4: Marsha Samuels, 55 Gladiola, San Antonio, TX 65925
5: Jackie Smith, 871 West Samuels, Orlando, FL 53290
6: Michael Taylor, 2234 Market Ave., Cincinnati, OH 44631
7: Bob Willis, 543 Grand Ave, Santa Barbara, CA 94007
8: F6
```

Press E Enter to save the file and exit to the system prompt.

You can also leave EDLIN without saving the data. Type Q Enter.

EDLIN displays the following prompt:

```
Abort edit (Y/N)?
```

Type Y to return to the system prompt or N to return to EDLIN.

12.2.3 Printing with EDLIN

Now you can print your address file. The simplest way to do this is to redirect the data to the printer with Ctrl and Prt Sc and then enter TYPE ADDRESS.TXT. The following is another method of printing. Make sure your printer is connected, switched on and filled with paper. Type the following and then press Enter:

```
COPY ADDRESS.TXT PRN
```

Summary

Typing EDLIN FILENAME starts the EDLIN program. The file can be new or may already exist. EDLIN.COM must be available on disk since EDLIN is a transient DOS command.

Typing I Enter allows you to write text to the file.

Files created by EDLIN can be printed using COPY FILENAME.EXT PRN.

You can leave EDLIN, without saving text, from the command prompt by typing Q Enter. EDLIN asks if you want to abort the edit. Pressing Y returns you to the DOS system prompt. Pressing N keeps you in EDLIN.

12.3 Special Capabilities Of EDLIN

This section shows that EDLIN can do much more than simply load, save and edit files. Eventually you'll probably use a true word processing application for writing and editing text files. EDLIN is very primitive compared to most word processors on the market, so we'll only briefly discuss EDLIN's capabilities.

12.3.1 Replacing characters and words with EDLIN

Let's reload and edit the ADDRESS.TXT file. Type the following and then press Enter:

```
EDLIN ADDRESS.TXT
```

Type the following and then press Enter:

```
L
```

The listing again appears:

```
1: Andrew Bailey, 325 Washington St, Salt Lake City, UT 75680
2: Jill Martin, 788 N. Elm St, San Diego, CA 94008
3: Bill Remington, 104 Ocean View, Santa Barbara, CA 94007
4: Marsha Samuels, 55 Gladiola, San Antonio, TX 65925
5: Jackie Smith, 871 West Samuels, Orlando, FL 53290
6: Michael Taylor, 2234 Market Ave., Cincinnati, OH 44631
7: Bob Willis, 543 Grand Ave, Santa Barbara, CA 94007
```

Read through the list. The street name in Jackie Smith's address should be "Samuel," not "Samuels". With EDLIN you can easily make the correction.

Type the following (press the keys as stated):

```
5,5?RSamuelsF6Samuel Enter
```

The system responds:

```
        5: Jackie Smith, 871 West Samuel, Orlando, FL, 53290
O.K.?
```

Press Y to get the command prompt. Type 5 L Enter to see the corrected line.

You can replace all occurrences in the file by typing the command to include all text lines in the file. For example, the following command changes all "Samuels" references in the text to "Samuel", which would make line 4 incorrect (don't type this):

```
1,7?RSamuels F6 Samuel Enter
```

If the two numbers at the beginning of the command are identical, only that line is changed, as you saw above.

After reading through your list you realize that, because she was recently married, Jill Martin's last name should be changed to Sanford. So type the following and then press Enter:

```
2,2?RMartin F6 Sanford
```

The name is changed to:

```
2: Jill Sanford, 788 N. Elm Street, San Diego, CA, 94008
```

12.3.2 Moving lines with EDLIN

Since you changed Jill's last name, the list is no longer in alphabetical order. EDLIN contains a command for moving lines.

Type the following and then press Enter:

```
2,2,5m
```

M   M (MOVE) moves a range of lines (2,2) to a new location (5). L Enter displays the following:

```
1: Andrew Bailey, 325 Washington St, Salt Lake City, UT 75680
2: Bill Remington, 104 Ocean View, Santa Barbara, CA 94007
3: Marsha Samuels, 55 Gladiola, San Antonio, TX 65925
4: Jill Sanford, 788 N. Elm St, San Diego, CA 94008
5: Jackie Smith, 871 West Samuel, Orlando, FL 53290
6: Michael Taylor, 2234 Market Ave., Cincinnati, OH 44631
7: Bob Willis, 543 Grand Ave., Santa Barbara, CA 94007
```

12.3.3 Copying lines with EDLIN

You can also copy a line. For example, if you have a long list of addresses and want to copy the addresses of people who should be the first to receive a postcard.

Type:

```
5,5,1c Enter
```

C This instructs EDLIN to copy a range of lines (5,5) up to line 1:

```
1: Jackie Smith, 871 West Samuel, Orlando, FL 53290
2: Andrew Bailey, 325 Washington St, Salt Lake City, UT 75680
3: Bill Remington, 104 Ocean View, Santa Barbara, CA 94007
4: Marsha Samuels, 55 Gladiola, San Antonio, TX 65925
5: Jill Sanford, 788 N. Elm St, San Diego, CA 94008
6: Jackie Smith, 871 West Samuel, Orlando, FL 53290
7: Michael Taylor, 2234 Market Ave., Cincinnati, OH, 44631
8: Bob Willis, 543 Grand Ave., Santa Barbara, CA 94007
```

Now you can add another name to the beginning. Type the following and then press Enter:

```
5,5,2c
```

After you press L Enter, the result is:

```
1: Jackie Smith, 871 West Samuel, Orlando, FL 53290
2: Jill Sanford, 788 N. Elm St, San Diego, CA 94008
3: Andrew Bailey, 325 Washington St, Salt Lake City, UT 75680
4: Bill Remington, 104 Ocean View, Santa Barbara, CA 94007
5: Marsha Samuels, 55 Gladiola, San Antonio, TX 65925
6: Jill Sanford, 788 N. Elm St, San Diego, CA 94008
7: Jackie Smith, 871 West Samuel, Orlando, FL 53290
8: Michael Taylor, 2234 Market Ave., Cincinnati, OH 44631
9: Bob Willis, 543 Grand Ave., Santa Barbara, CA 94007
```

To separate the lines that were taken out of the alphabetical list, insert a line consisting of equal signs. Type the following and then press Enter after each line:

```
3I
=======================================================
F6
```

Now the file looks like this:

```
 1: Jackie Smith, 871 West Samuel, Orlando, FL 53290
 2: Jill Sanford, 788 N. Elm St, San Diego, CA 94008
 3: ========================================================
 4: Andrew Bailey, 325 Washington St, Salt Lake City, UT 75680
 5: Bill Remington, 104 Ocean View, Santa Barbara, CA 94007
 6: Marsha Samuels, 55 Gladiola, San Antonio, TX 65925
 7: Jill Sanford, 788 N. Elm St, San Diego, CA 94008
 8: Jackie Smith, 871 West Samuel, Orlando, FL 53290
 9: Michael Taylor, 2234 Market Ave., Cincinnati, OH 44631
10: Bob Willis, 543 Grand Ave., Santa Barbara, CA 94007
```

Press E Enter to save the file and exit EDLIN.

12.3.4 Displaying and deleting lines

Finally, here are some hints to use if your address list becomes too long. If you want to display single lines, you can type:

```
5,9L
```

Then press Enter. This instructs EDLIN to display lines 5 through 9.

The D command could be used to delete multiple lines:

```
11,15D
```

Then press Enter.

12.4 Review

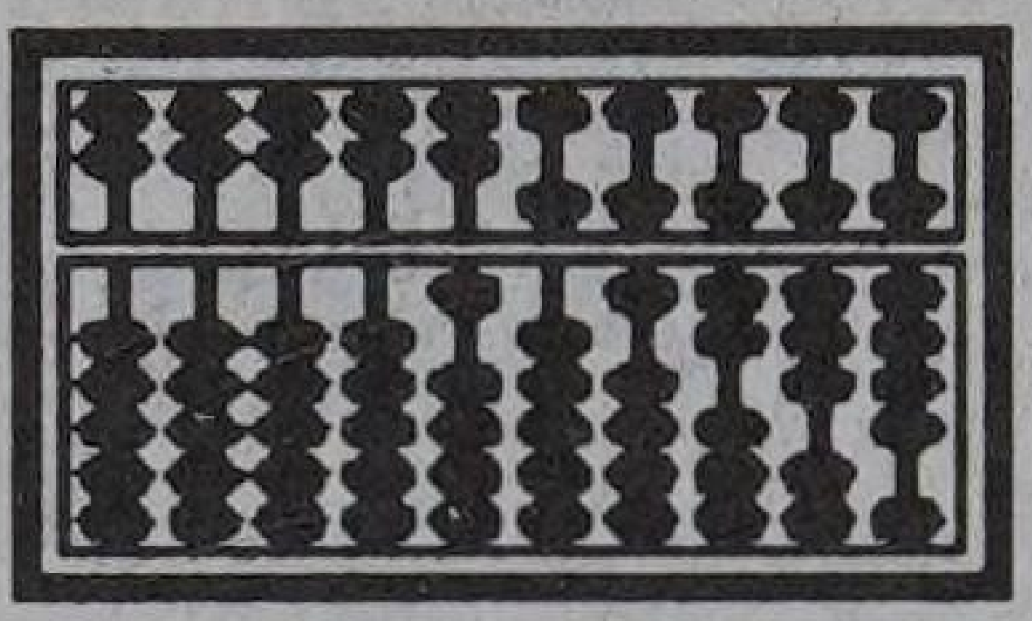
Here's what we learned about EDLIN:

EDLIN is a simple line editor, which can be used for entering, saving and editing text files.

The EDLIN command set covered in this chapter consists of the following:

C	Copies lines from one location in a file to another.
D	Deletes lines.
E	Ends edit, saves file and returns to DOS.
I	Inserts lines.
L	Lists lines.
M	Moves lines from one location in a file to another.
Q	Quits EDLIN without saving edit.
R	Replaces one text with another, either in individual lines, ranges of lines or entire files.

Chapter



13

Using The MS-DOS Editor

The MS-DOS Editor is a full screen editor that lets you create, view and change text files quickly and easily. It's comparable to a simple word processor. You can view a file on the full screen and use the cursor keys to easily change the contents of the file.

Fundamentals

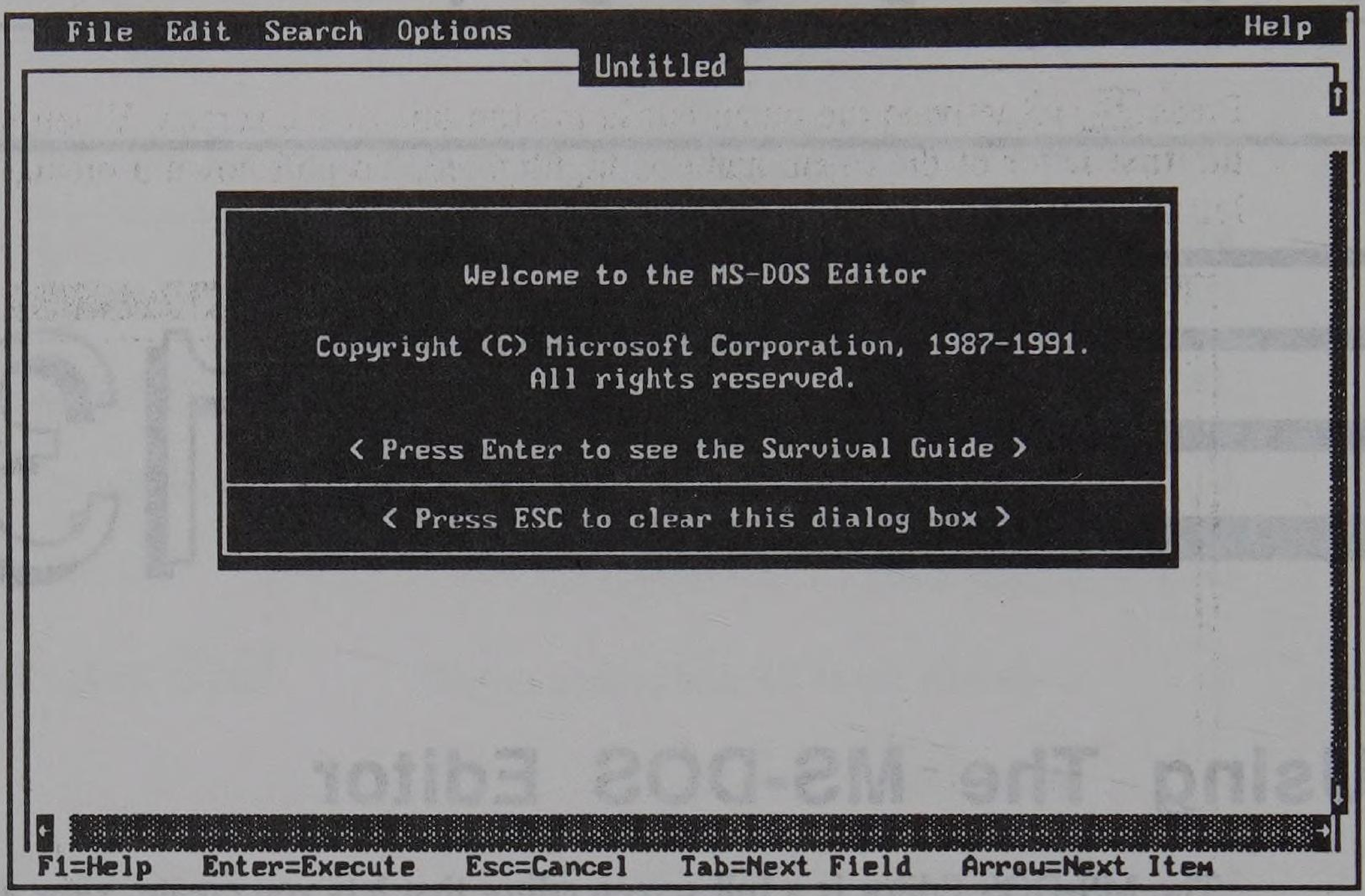
Like the MS-DOS Shell and QBasic, MS-DOS Editor is based on the SAA (System Application Architecture) standard. The editing features of MS-DOS Editor can be used by pressing appropriate keys on the keyboard or by using a mouse.

In this chapter we'll show you how to make your selections at the keyboard. Of course, you can also select the same functions by clicking on the menus with the mouse by pointing and clicking on the appropriate commands.

To start the MS-DOS Editor, type the following and then press Enter:

```
EDIT
```

The following screen appears:


File Edit Search Options
Help
Untitled
F1=Help Enter=Execute Esc=Cancel Tab=Next Field Arrow=Next Item

Welcome to the MS-DOS Editor

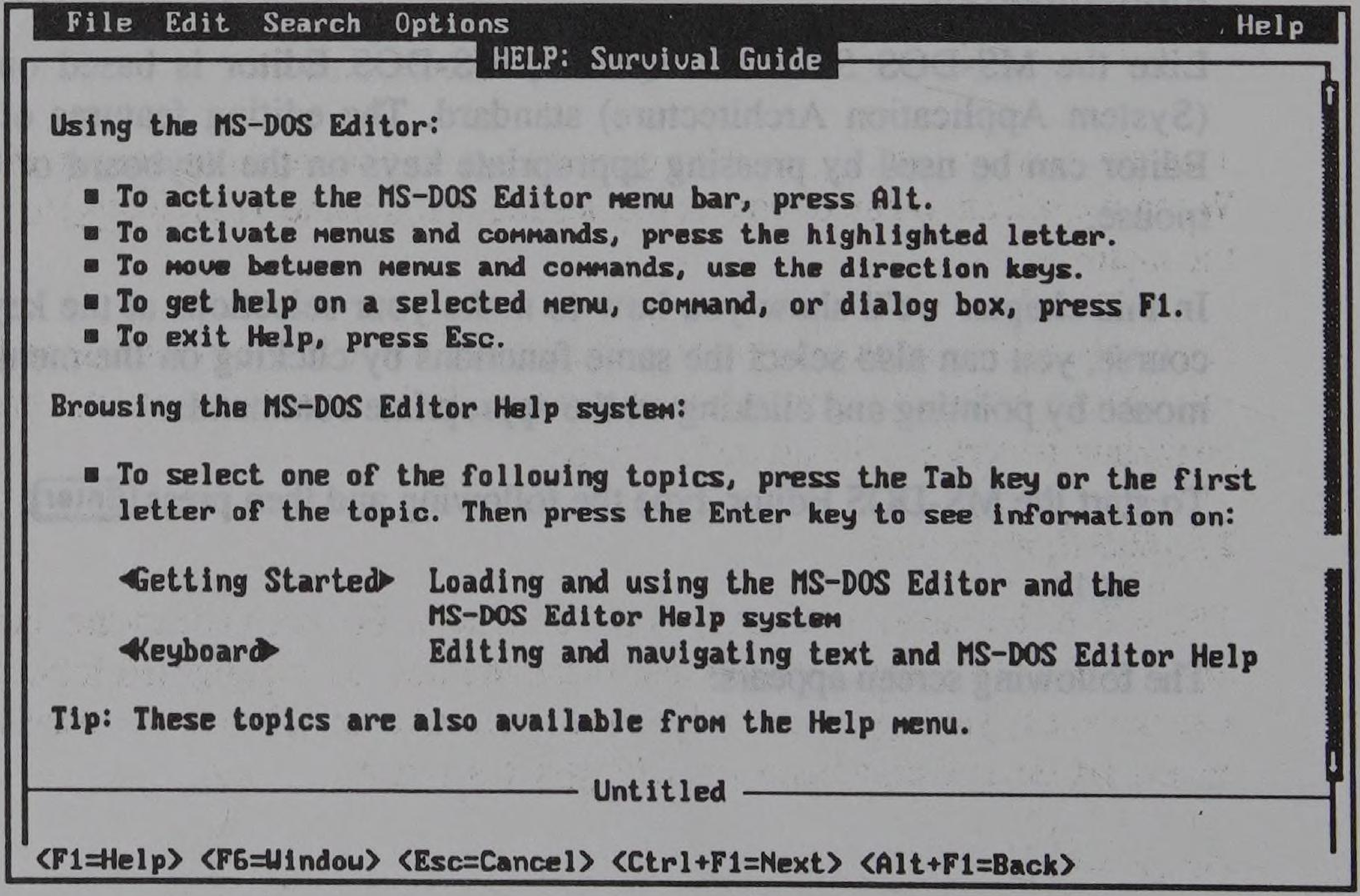
Copyright (C) Microsoft Corporation, 1987-1991.
All rights reserved.

< Press Enter to see the Survival Guide >

< Press ESC to clear this dialog box >

MS-DOS Editor - the full screen editor

You can receive information on the MS-DOS Editor by pressing Enter. This opens the Survival Guide help files:


File Edit Search Options
Help
HELP: Survival Guide
Untitled
<F1=Help> <F6=Window> <Esc=Cancel> <Ctrl+F1=Next> <Alt+F1=Back>

Using the MS-DOS Editor:

- To activate the MS-DOS Editor menu bar, press Alt.
- To activate menus and commands, press the highlighted letter.
- To move between menus and commands, use the direction keys.
- To get help on a selected menu, command, or dialog box, press F1.
- To exit Help, press Esc.

Browsing the MS-DOS Editor Help system:

- To select one of the following topics, press the Tab key or the first letter of the topic. Then press the Enter key to see information on:

◄Getting Started►	Loading and using the MS-DOS Editor and the MS-DOS Editor Help system
◄Keyboard►	Editing and navigating text and MS-DOS Editor Help

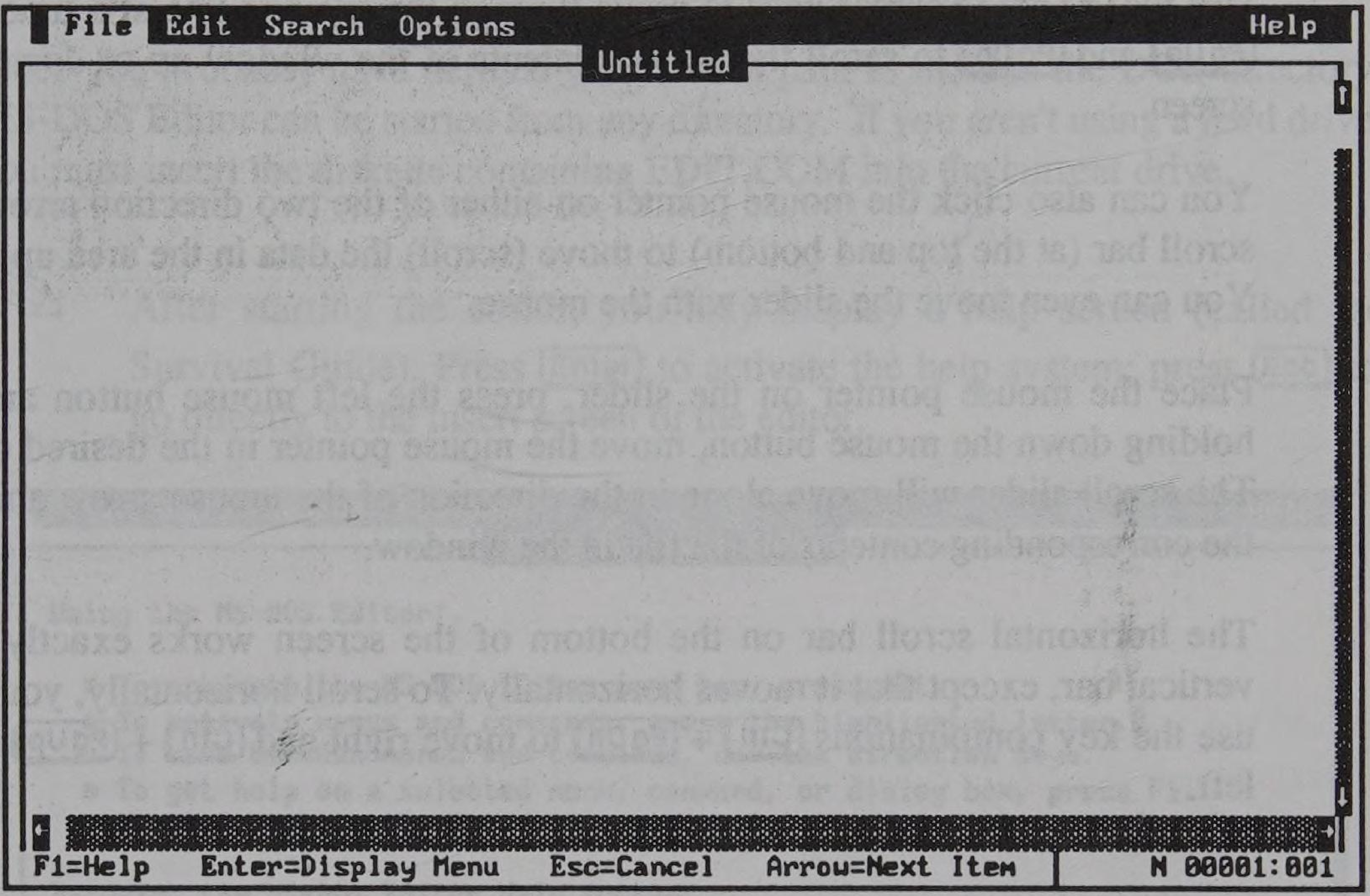
Tip: These topics are also available from the Help menu.

The MS-DOS Editor help files

Alternatively, you can press Esc to clear the dialog box.

The menu bar

Press Alt to activate the menu bar in the top line of the screen. When it's active, the first letter of the menu name is highlighted. To pull down a menu, press the letter corresponding to the menu name.



The Edit menu bar activated

You can also use the → and ← cursor keys to choose from among the menus. When the selection cursor covers the desired menu name, press Enter to pull down the menu.

If you are using the mouse, point to the desired name in the menu bar and click the left mouse button to pull down the menu. Point and click on the appropriate command to perform the desired function.

The pull-down menus

The pull-down menus contain lists of commands. For each command, one letter is highlighted. To select a command, press the desired highlighted letter. You can also use the ↑ and ↓ cursor keys to select a command. When the selection cursor covers the desired command, press Enter to select it.

The scroll bar

On the right margin of the screen is a vertical grey strip with arrows pointing up and down, as seen in the previous figure. This is the scroll bar. Within this strip is a rectangle. This is the scroll slider.

Its position within the scroll bar indicates which section of the file is being displayed. If the slider is at the top of the scroll bar, the start of the file is being displayed. If it is in the middle of the scroll bar, you are seeing the middle portion and if it's at the bottom, the window contains the last part of the file.

Use the ↑ or ↓ cursor keys to move through the file one line at a time or press PgUp and PgDn to scroll the entire contents of the window up or down by one screen.

You can also click the mouse pointer on either of the two direction arrows of the scroll bar (at the top and bottom) to move (scroll) the data in the area up or down. You can even move the slider with the mouse.

Place the mouse pointer on the slider, press the left mouse button and, while holding down the mouse button, move the mouse pointer in the desired direction. The scroll slider will move along in the direction of the mouse arrow and display the corresponding contents of the file in the window.

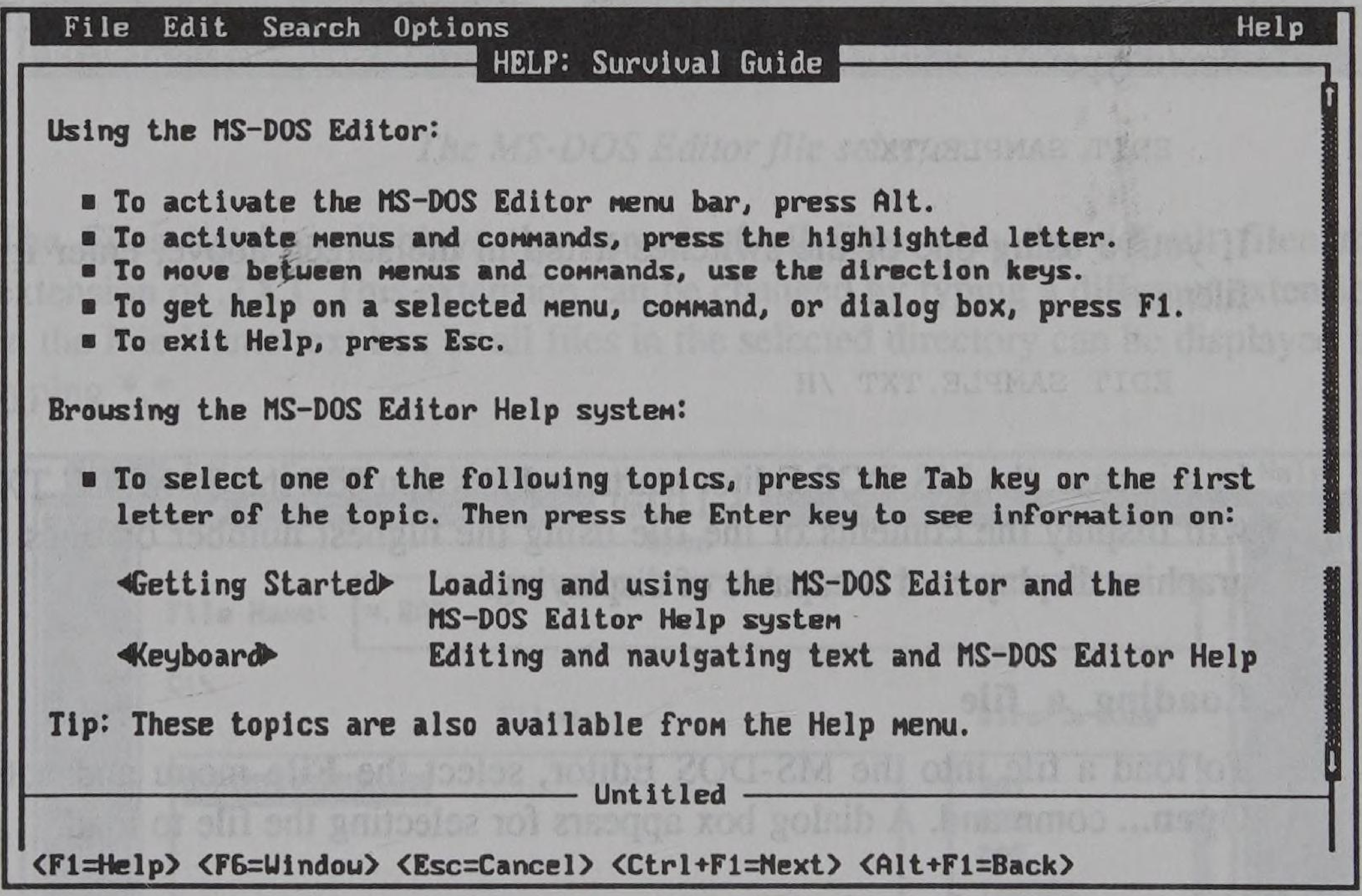
The horizontal scroll bar on the bottom of the screen works exactly like the vertical bar, except that it moves horizontally. To scroll horizontally, you can also use the key combinations Ctrl + PgDn to move right and Ctrl + PgUp to move left.

13.1 Starting The Editor And Loading A File

Since the MS-DOS Editor is a transient program, you must load it from your diskette or hard drive every time. When DOS 5.0 is installed, the EDIT.COM program is installed in the DOS directory.

Since you probably have defined your search path to include the DOS directory, MS-DOS Editor can be started from any directory. If you aren't using a hard drive, you must insert the diskette containing EDIT.COM into the current drive.

After starting the editor, you may display a help screen (called the Survival Guide). Press Enter to activate the help system; press Esc to go directly to the insert screen of the editor.


File
Edit
Search
Options
Help
HELP: Survival Guide
Untitled
<F1=Help> <F6=Window> <Esc=Cancel> <Ctrl+F1=Next> <Alt+F1=Back>

```
Using the MS-DOS Editor:

  ■ To activate the MS-DOS Editor menu bar, press Alt.
  ■ To activate menus and commands, press the highlighted letter.
  ■ To move between menus and commands, use the direction keys.
  ■ To get help on a selected menu, command, or dialog box, press F1.
  ■ To exit Help, press Esc.

Browsing the MS-DOS Editor Help system:

  ■ To select one of the following topics, press the Tab key or the first
    letter of the topic. Then press the Enter key to see information on:

    ◄Getting Started►  Loading and using the MS-DOS Editor and the
                       MS-DOS Editor Help system
    ◄Keyboard►         Editing and navigating text and MS-DOS Editor Help

Tip: These topics are also available from the Help menu.
```

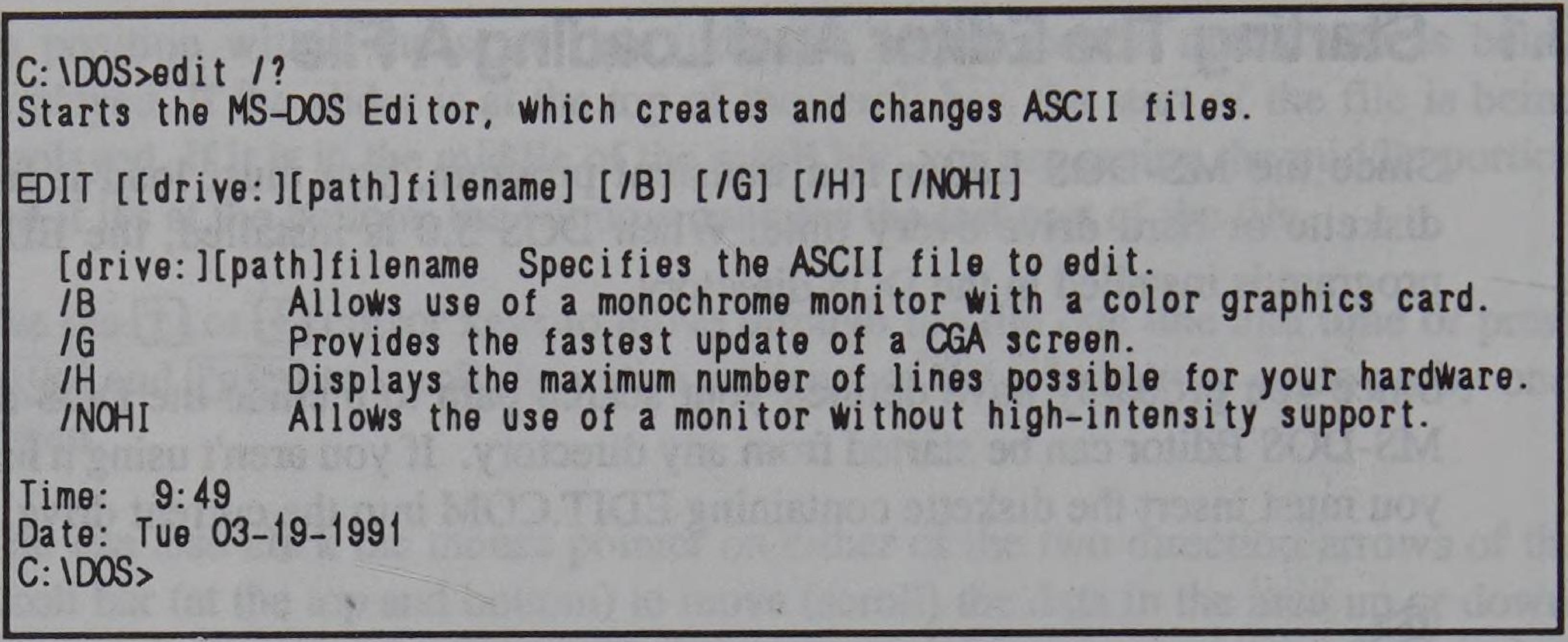
The MS-DOS Editor help system

Starting the MS-DOS Editor

Type the EDIT command to start the editor.

As with all resident and external MS-DOS commands of Version 5.0, when you call the MS-DOS Editor you can specify the /? switch to display a condensed help screen explaining the command syntax.

This help screen displays the various options available when starting the editor.



```
C:\DOS>edit /?
Starts the MS-DOS Editor, which creates and changes ASCII files.

EDIT [[drive:][path]filename] [/B] [/G] [/H] [/NOHI]

  [drive:][path]filename  Specifies the ASCII file to edit.
  /B          Allows use of a monochrome monitor with a color graphics card.
  /G          Provides the fastest update of a CGA screen.
  /H          Displays the maximum number of lines possible for your hardware.
  /NOHI       Allows the use of a monitor without high-intensity support.

Time:  9:49
Date: Tue 03-19-1991
C:\DOS>
```

Entering EDIT /? from the command line

As you can see from the syntax description, you can run the MS-DOS Editor to edit a specific file. For example, to edit an ASCII file named SAMPLE.TXT, you would type:

```
EDIT SAMPLE.TXT
```

If you're using one of the switches listed in the screen above, enter it after the filename:

```
EDIT SAMPLE.TXT /H
```

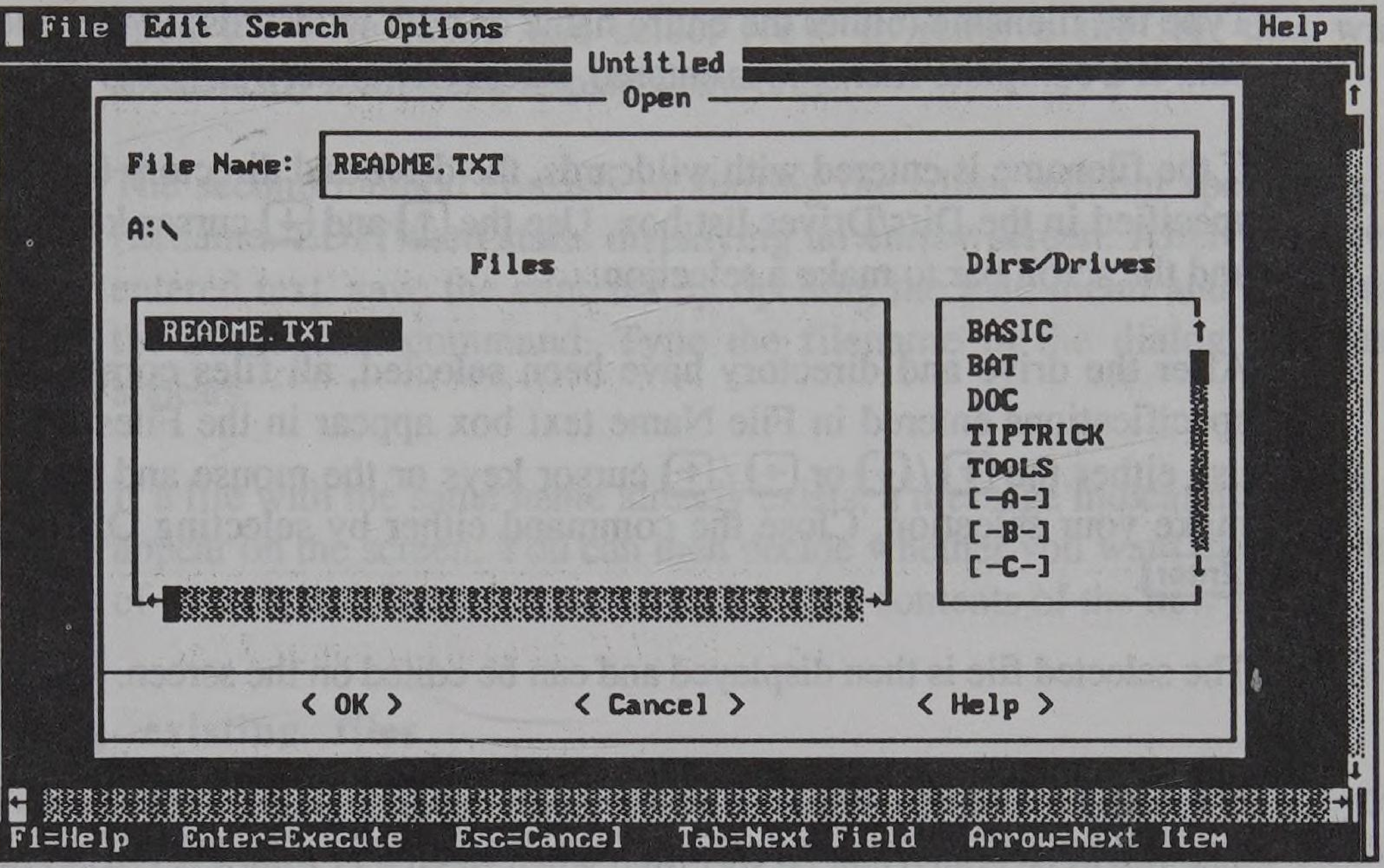
In this case, the MS-DOS Editor starts and lets you edit the SAMPLE.TXT file. It will display the contents of the file using the highest number of lines that your graphics display card is capable of displaying.

Loading a file

To load a file into the MS-DOS Editor, select the **File** menu and activate the **Open...** command. A dialog box appears for selecting the file to load.

You can select from the File Name: text box, two list boxes (Files and Dirs/Drives) and three command buttons (OK, Cancel and Help).

Use the Tab key to move between the boxes and command buttons. The mouse pointer can also be used to select the text box or list boxes.


File
Edit
Search
Options
Help
Untitled
Open
Files
Dirs/Drives
< OK >
< Cancel >
< Help >
F1=Help
Enter=Execute
Esc=Cancel
Tab=Next Field
Arrow=Next Item

File Name: README.TXT

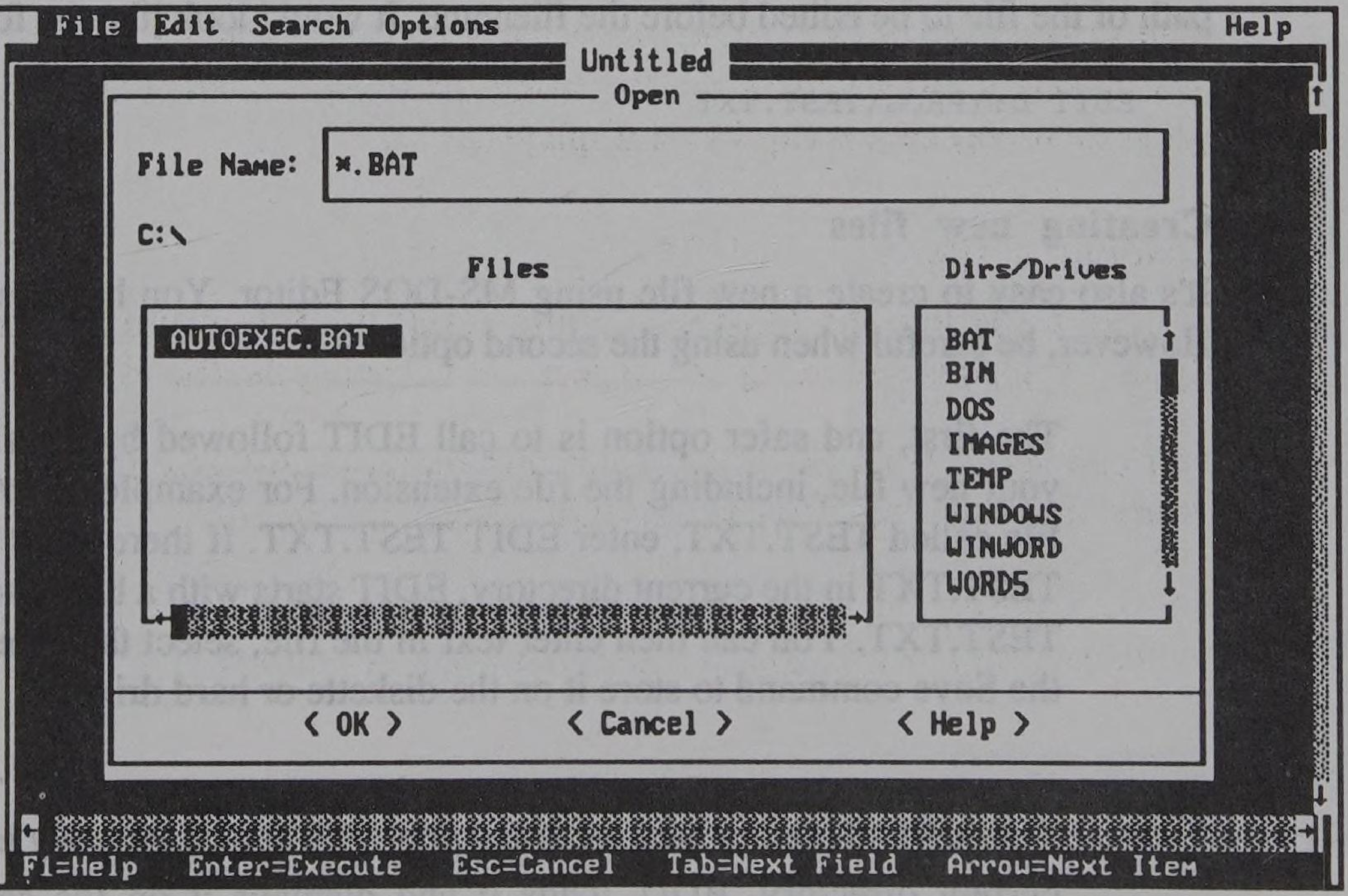
A:\

README.TXT

BASIC
BAT
DOC
TIPTRICK
TOOLS
[-A-]
[-B-]
[-C-]

The MS-DOS Editor file selector

The Files list box displays the names of all files with the default filename extension of .TXT. This extension can be changed by typing a different extension in the File Name text box or all files in the selected directory can be displayed by typing *.*.


File
Edit
Search
Options
Help
Untitled
Open
Files
Dirs/Drives
< OK >
< Cancel >
< Help >
F1=Help
Enter=Execute
Esc=Cancel
Tab=Next Field
Arrow=Next Item

File Name: *.BAT

C:\

AUTOEXEC.BAT

BAT
BIN
DOS
IMAGES
TEMP
WINDOWS
WINWORD
WORD5

*MS-DOS Editor displaying *.BAT files in the root directory of drive C:*

Type the filename (either the entire name or with wildcards) in the File Name text box. If a complete filename is entered, select OK; otherwise press Enter.

If the filename is entered with wildcards, the drive and directory for the file can be specified in the Dirs/Drives list box. Use the ↑ and ↓ cursor keys or the mouse and the scroll bar to make a selection.

After the drive and directory have been selected, all files corresponding to the specifications entered in File Name text box appear in the Files list box. Again, use either the ↑ / ↓ or → / ← cursor keys or the mouse and the scroll bar to make your selection. Close the command either by selecting OK or by pressing Enter.

The selected file is then displayed and can be edited on the screen.

Starting with a file

The file to be edited can be specified when MS-DOS Editor is started by typing:

```
EDIT filename
```

MS-DOS Editor then starts and the contents of the file are displayed on the screen. If the EDIT.COM file is in a defined path, then you can change to the directory containing the file to be edited and start MS-DOS Editor. Alternatively, you can start MS-DOS Editor with a file from any directory by specifying the complete path of the file to be edited before the filename. It would look like the following:

```
EDIT D:\PROG\TEST.TXT
```

Creating new files

It's also easy to create a new file using MS-DOS Editor. You have two options. However, be careful when using the second option.

The first, and safer option is to call EDIT followed by the filename of your new file, including the file extension. For example, to create a new file called TEST.TXT, enter EDIT TEST.TXT. If there is no file named TEST.TXT in the current directory, EDIT starts with a blank screen titled TEST.TXT. You can then enter text in the file, select the **File** menu and the **Save** command to store it on the diskette or hard drive.

This method is safer because if this file (TEST.TXT) already exists in the current directory, EDIT loads it and displays it on the screen. This reminds you that the filename already exists and that you will need to select a different name for your new file. (If you accidentally load a file, open the **File** menu and select **New** to remove the file that was accidentally loaded and display a new area.)

The second method consists of starting the editor without specifying a filename. EDIT then starts displaying an empty screen. After you have entered text, save the contents by opening the **File** menu and selecting the **Save As...** command. Type the filename in the dialog box that appears.

If a file with the same name already exists, a message indicating this will appear on the screen. You can then decide whether you want the contents of the original file to be overwritten by the contents of the new file.

Editing existing files

As mentioned above, an existing file can be loaded when starting the editor or you can use the **Open...** command from the **File** menu. The file is displayed on the screen for editing.

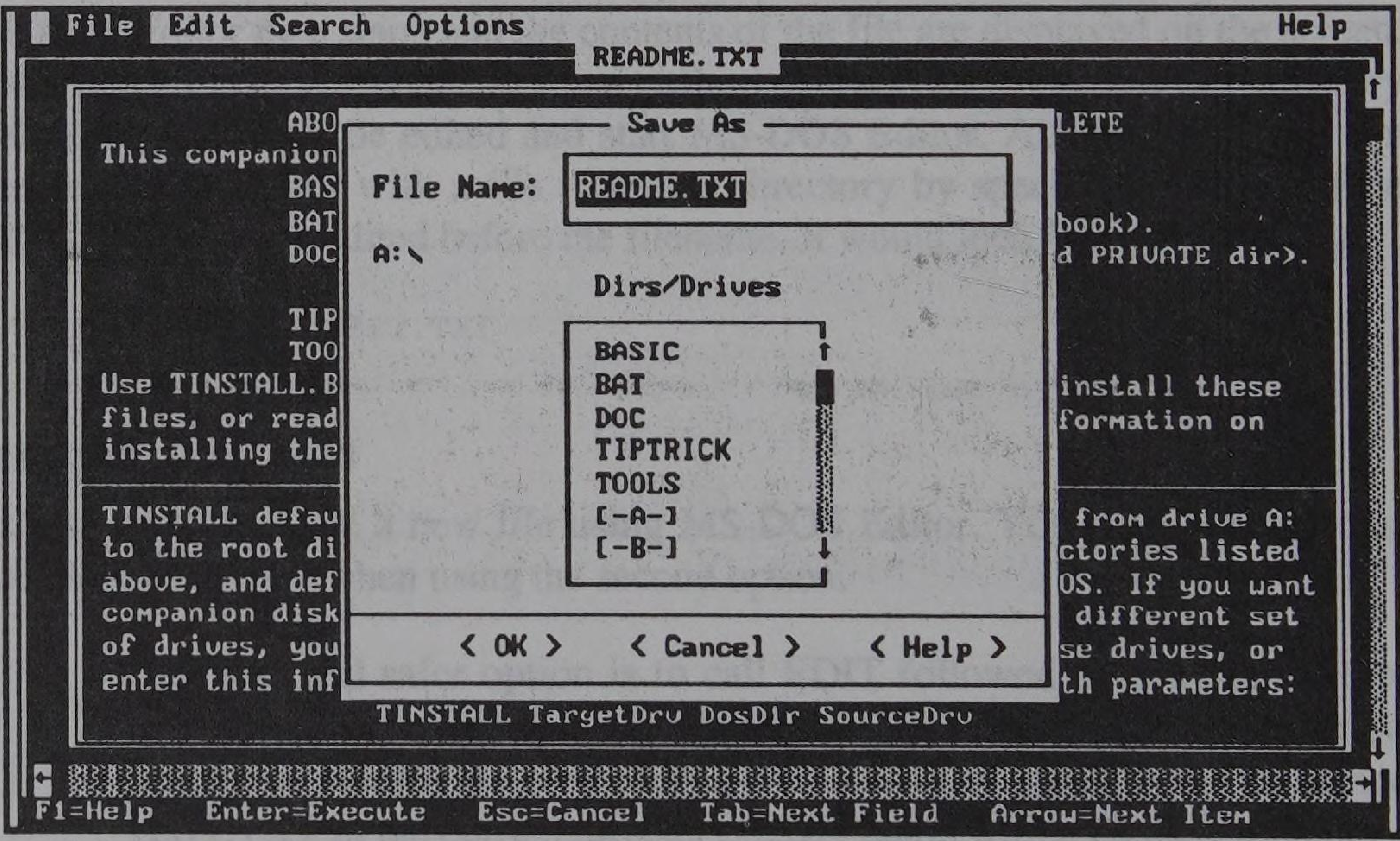
13.2 Saving A File

One of the most important functions of any editor is saving files. The MS-DOS Editor saves data in ASCII format. You'll be able to use many other editors or word processing programs with text files created by MS-DOS Editor.

Remember that when you save a file in the MS-DOS Editor, a new file will overwrite a file with the same name that already exists in the current directory of the hard drive or diskette. The MS-DOS Editor doesn't make a backup copy of the old file. This means that there isn't a .BAK file containing the contents of the previous version of the file. By overwriting an earlier file with the contents of the new file, you will lose the data.

Saving a new file

To save a new file that doesn't have a name, open the **File** menu and select the **Save As...** command. The Save As dialog box appears with the File Name text box and Dirs/Drives list box. Select the drive and directory, in which the file should be saved, from the Dirs/Drives list box. Type the filename in the File Name text box. Finally, press Enter or select OK to save the file.



File Name: README.TXT

A:\

File
Edit
Search
Options
Help
README.TXT
Save As
Dirs/Drives
BASIC
BAT
DOC
TIPTRICK
TOOLS
[-A-]
[-B-]
< OK >
< Cancel >
< Help >
ABO
This companion
BAS
BAT
DOC
TIP
TOO
Use TINSTALL.B
files, or read
installing the
TINSTALL defau
to the root di
above, and def
companion disk
of drives, you
enter this inf
LETE
book).
d PRIVATE dir>.
install these
formation on
from drive A:
ctories listed
OS. If you want
different set
se drives, or
th parameters:
TINSTALL TargetDrv DosDir SourceDrv
F1=Help
Enter=Execute
Esc=Cancel
Tab=Next Field
Arrow=Next Item

The MS-DOS Editor Save As dialog box

Saving an open file

If you open and edit an existing file or you start EDIT with a filename, the filename and its directory are already established. You can save the file by using the **Save** command from the **File** menu.

You can also change the filename and directory before saving the file. Just open the **File** menu and select the **Save As...** command. Type the filename in the

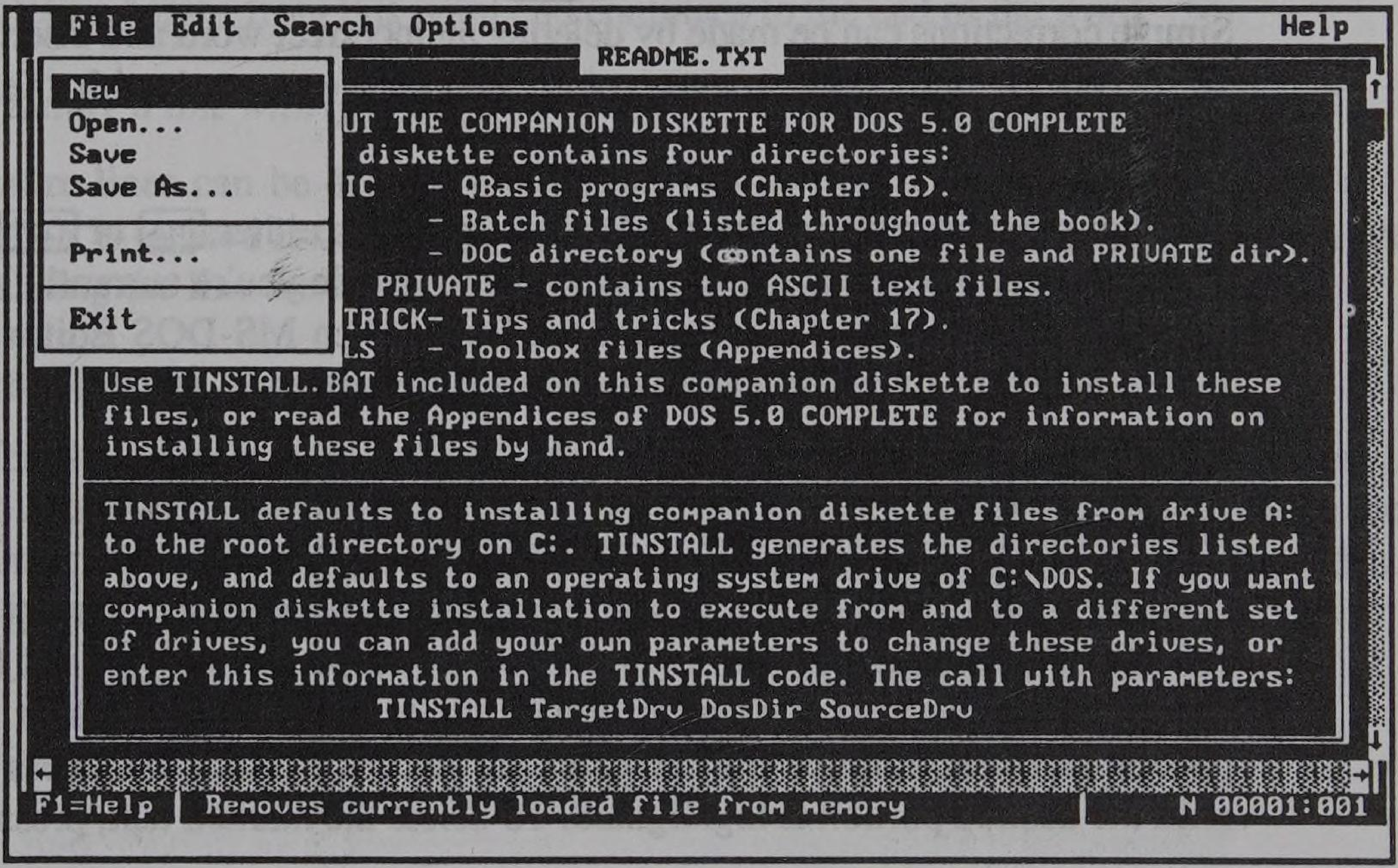
File Name text box. The directory from which the file was loaded or the active directory is selected from the Dirs/Drives list box beneath.

Changing filenames

You can also use the **Save As...** command to save the file with a new name. Type the new filename and press Enter or select OK. The MS-DOS Editor then stores the file on your hard drive or diskette using the new filename and preserving the original file.

Clearing the screen and the main memory

After saving a file, the contents of the file are still displayed on the screen and remain in memory. To create a new file, use the **New** command from the **File** menu. When you activate this command, it clears the contents of the file from memory and the screen.



```
UT THE COMPANION DISKETTE FOR DOS 5.0 COMPLETE
 diskette contains four directories:
IC    - QBasic programs (Chapter 16).
      - Batch files (listed throughout the book).
      - DOC directory (contains one file and PRIVATE dir).
  PRIVATE - contains two ASCII text files.
TRICK- Tips and tricks (Chapter 17).
LS    - Toolbox files (Appendices).
Use TINSTALL.BAT included on this companion diskette to install these
files, or read the Appendices of DOS 5.0 COMPLETE for information on
installing these files by hand.

TINSTALL defaults to installing companion diskette files from drive A:
to the root directory on C:. TINSTALL generates the directories listed
above, and defaults to an operating system drive of C:\DOS. If you want
companion diskette installation to execute from and to a different set
of drives, you can add your own parameters to change these drives, or
enter this information in the TINSTALL code. The call with parameters:
                TINSTALL TargetDrv DosDir SourceDrv
F1=Help | Removes currently loaded file from memory          N 00001:001
```

Selecting New from the File menu

Quitting MS-DOS Editor without saving

To quit the editor without saving the file in the memory, select the **Exit** command in the **File** menu.

If you made any changes to the file since the last save, MS-DOS Editor will remind you that you haven't yet saved the contents of the file to disk. Select Yes to save. If you choose No, the file is not saved and you leave the editor.

13.3 Entering And Editing Text

The full screen editor makes it easy to enter text and edit existing text.

Entering text

Use the keyboard to enter text in the editor just as you would use a typewriter. Unlike word processing programs, MS-DOS Editor doesn't automatically make a new line when you reach the end of a line (at 78 characters) on the screen. Instead, it shifts the displayed area farther to the right. The length of a text line is limited to 256 characters. If you reach the end of a line when entering text (at character 256), you'll hear a short warning beep. Press Enter to move to the beginning of the next line.

Inserting and overtyping text

Simple corrections can be made by deleting an incorrect word and inserting the new word (insert mode) or by overtyping the incorrect word with the new word (overtype mode).

Switching between these two modes is easy. Press either Ins or Ctrl + V. The appearance of the cursor will indicate in which mode you're currently located. The cursor appears as a line underneath the letter when MS-DOS Editor is in insert mode. When in overtype mode, the cursor changes to a rectangle the size of a letter.

Making changes in insert mode

In insert mode, first delete the word you want to replace. Do this by positioning the cursor at the start of the incorrect word and pressing Del until the word is deleted. You can also mark a portion of the text and then delete it. To mark a portion of the text, move the cursor to the start of the text and press Shift + → until the desired portion is highlighted. To delete the marked text, press Del.

Making changes in overtype mode

In overtype mode, a character covered by the blinking rectangle is replaced by a new typed character. After each keystroke, the cursor moves one character to the right. When using overtype mode, make sure that the text you're entering doesn't overtype any text that should remain. In case the new text entry is longer than the original text, press Ins (or Ctrl + V) to switch off overtype mode and switch on insert mode.

Deleting parts of text

There are several ways to delete passages of text, ranging from deleting letters, to words, or even entire lines.

Deleting a character with Backspace

You can use either the Backspace key or press Ctrl + H to delete a character to the left of the cursor. It doesn't matter whether you are in insert mode or overtype mode.

Deleting a character with Del

Use Del or press Ctrl + G to remove the character above the cursor (insert mode) or the character covered by the cursor (overtype mode).

Deleting a word with Ctrl+T

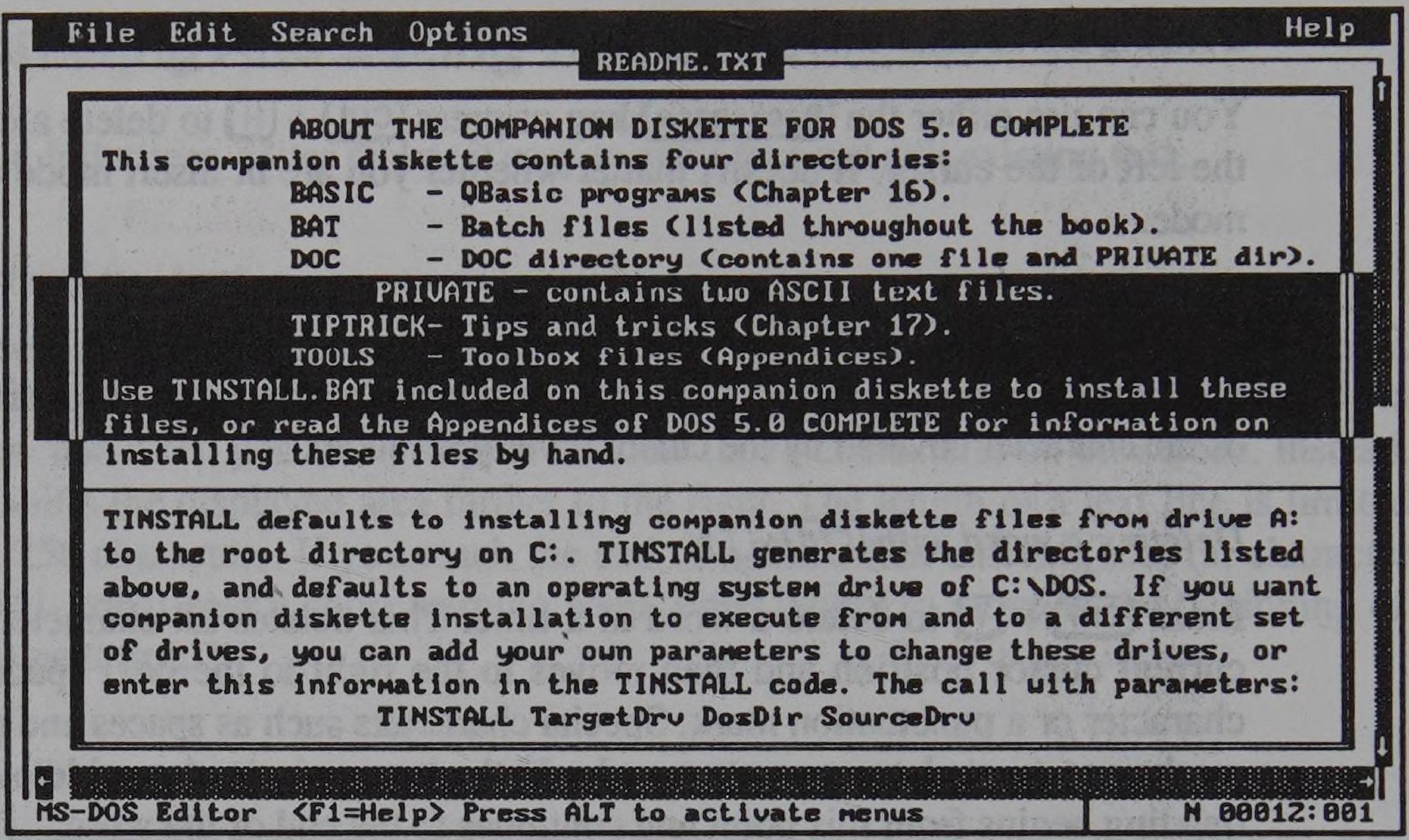
Press Ctrl + T to delete a word at a time. This deletes all characters from the current cursor position and then moves to the right to the next space, a special character or a punctuation mark. Special characters such as spaces and punctuation marks are treated as separate words. If the cursor is in the middle of a word, deleting begins from this point and continues to the end of the word.

Deleting a line with Ctrl+Y

Entire lines can be deleted by pressing Ctrl + Y. The entire line in which the cursor is located is deleted. It doesn't matter where the cursor is in the line. After deleting the line the cursor is positioned at the beginning of the next line.

Deleting a block using Del *to delete marked text*

Here's an easy way to delete blocks of text. Mark the block of text to be deleted, hold down the Shift key and press the cursor keys (→, ←, ↑ or ↓) to move the cursor; then press Del. MS-DOS Editor displays marked text in inverse video. After marking it, just press Del.



```
            ABOUT THE COMPANION DISKETTE FOR DOS 5.0 COMPLETE
This companion diskette contains four directories:
           BASIC   - QBasic programs (Chapter 16).
           BAT     - Batch files (listed throughout the book).
           DOC     - DOC directory (contains one file and PRIVATE dir).
                PRIVATE - contains two ASCII text files.
           TIPTRICK- Tips and tricks (Chapter 17).
           TOOLS   - Toolbox files (Appendices).
Use TINSTALL.BAT included on this companion diskette to install these
files, or read the Appendices of DOS 5.0 COMPLETE for information on
installing these files by hand.

TINSTALL defaults to installing companion diskette files from drive A:
to the root directory on C:. TINSTALL generates the directories listed
above, and defaults to an operating system drive of C:\DOS. If you want
companion diskette installation to execute from and to a different set
of drives, you can add your own parameters to change these drives, or
enter this information in the TINSTALL code. The call with parameters:
              TINSTALL TargetDrv DosDir SourceDrv
```

```
MS-DOS Editor  <F1=Help> Press ALT to activate menus            N 00012:001
```

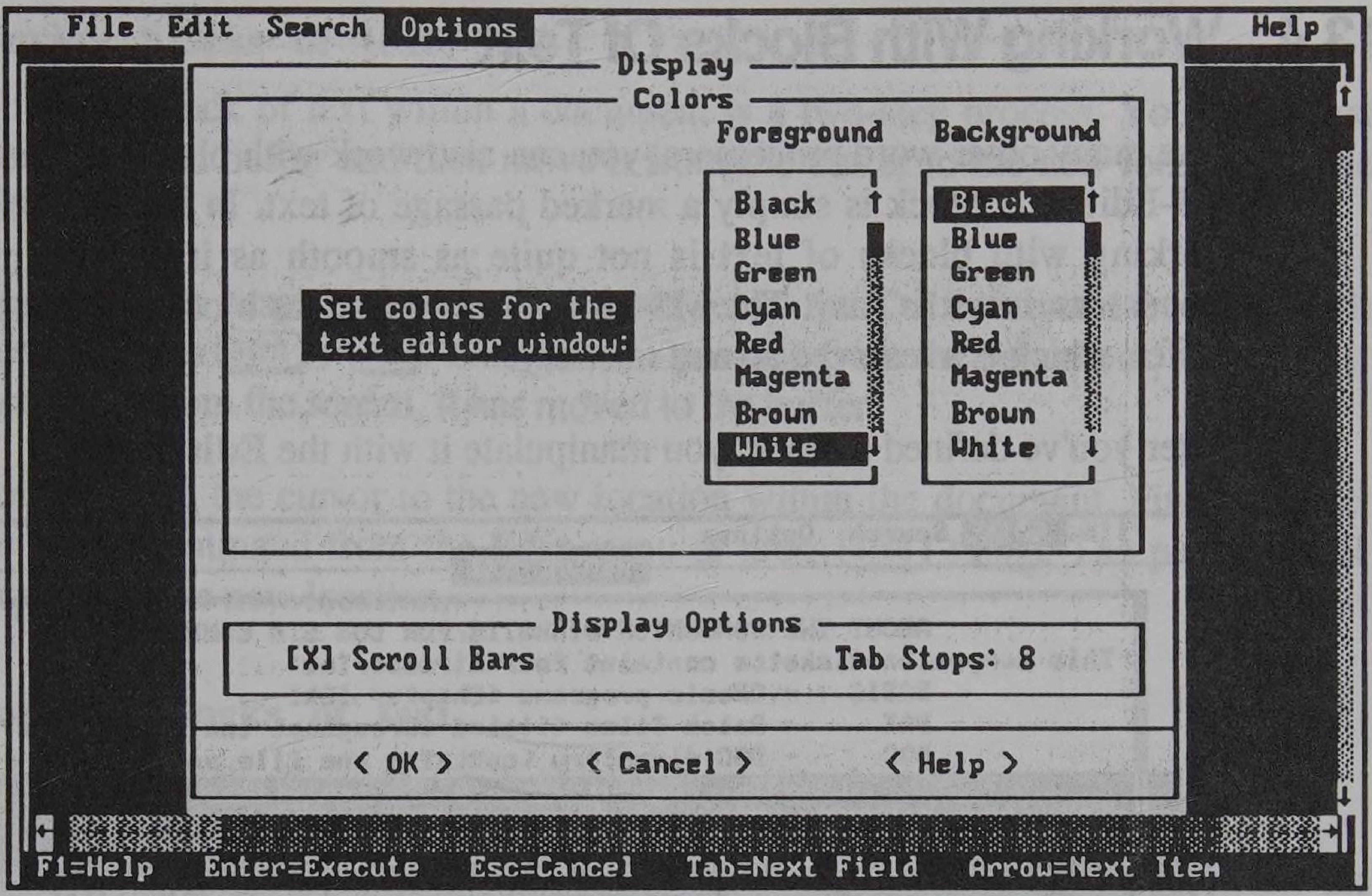
Marking a block of text

Tabs

You can also use tabs in the MS-DOS Editor. Tabs are imaginary stops on the screen. The cursor can be moved to these positions by pressing the Tab key. Tabs can be used to place columns of numbers underneath each other or set up tables.

The MS-DOS Editor doesn't insert ASCII tab characters; instead it inserts the corresponding number of spaces.

Before loading a file, containing tab characters, from a diskette or a hard drive, set the desired number of tab spaces. Do this using the **Display...** command from the **Options** menu. In the Display Options dialog box set the number of spaces between tabs by changing the Tab Stops option. The default setting is 8. Any value between 1 and 99 can be selected.


File Edit Search Options Help
Display
Colors
Foreground
Background
Black
Blue
Green
Cyan
Red
Magenta
Brown
White
Display Options
< OK >
< Cancel >
< Help >
F1=Help Enter=Execute Esc=Cancel Tab=Next Field Arrow=Next Item

Set colors for the text editor window:

Foreground: White

Background: Black

[X] Scroll Bars

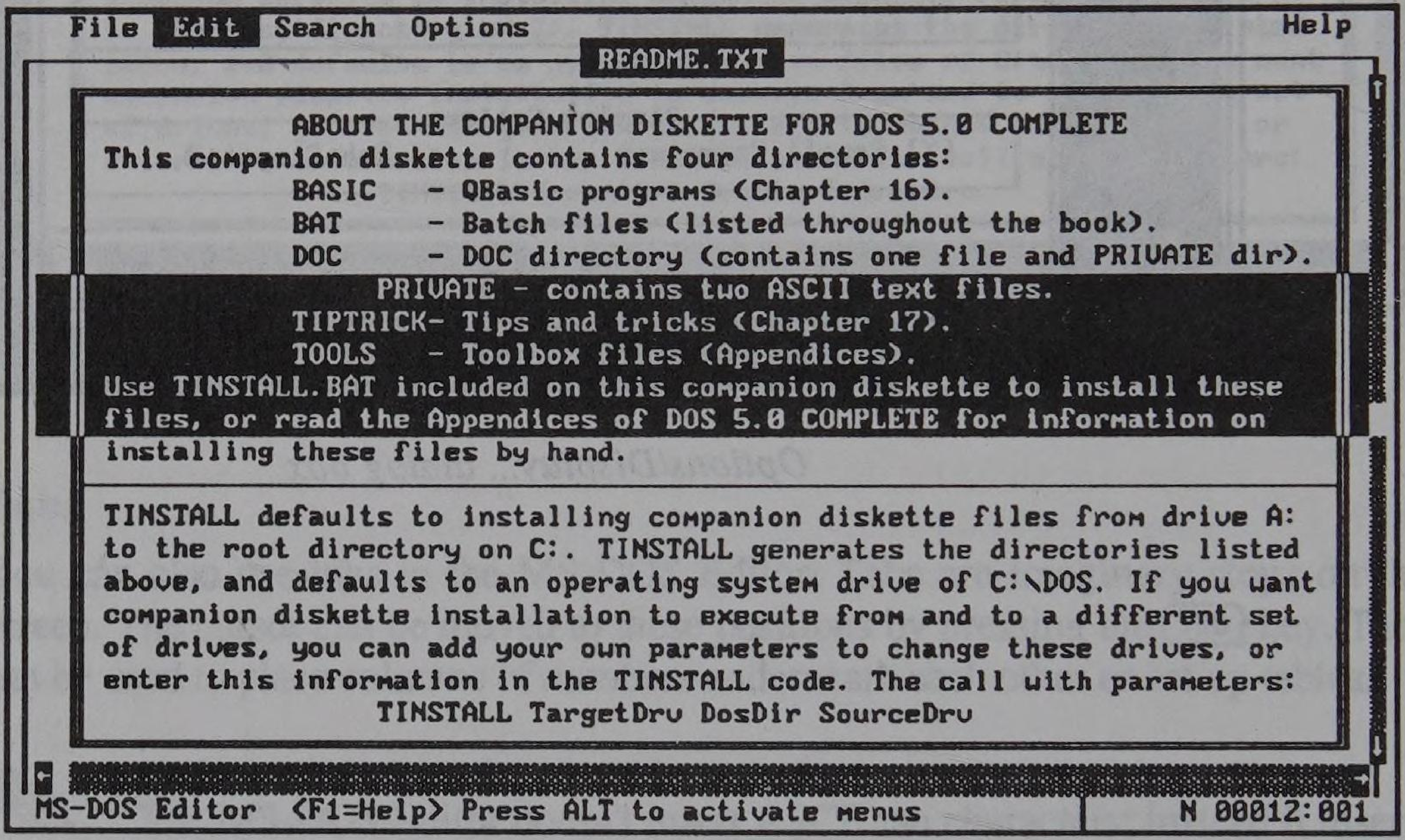
Tab Stops: 8

Options/Display... dialog box

13.4 Working With Blocks Of Text

Like many other word processors, you can also work with blocks of text using the MS-Editor. A block is simply a marked passage of text. In the MS-DOS Editor, working with blocks of text is not quite as smooth as it is with most word processors, but it is easy. The MS-DOS Editor manages blocks of text by using a buffer, which is a reserved area of memory.

After you've defined a block, you manipulate it with the **Edit** menu.



```
  File  Edit  Search  Options                                    Help
                          README.TXT
      ABOUT THE COMPANION DISKETTE FOR DOS 5.0 COMPLETE
This companion diskette contains four directories:
      BASIC    - QBasic programs (Chapter 16).
      BAT      - Batch files (listed throughout the book).
      DOC      - DOC directory (contains one file and PRIVATE dir).
               PRIVATE - contains two ASCII text files.
      TIPTRICK- Tips and tricks (Chapter 17).
      TOOLS    - Toolbox files (Appendices).
Use TINSTALL.BAT included on this companion diskette to install these
files, or read the Appendices of DOS 5.0 COMPLETE for information on
installing these files by hand.

TINSTALL defaults to installing companion diskette files from drive A:
to the root directory on C:. TINSTALL generates the directories listed
above, and defaults to an operating system drive of C:\DOS. If you want
companion diskette installation to execute from and to a different set
of drives, you can add your own parameters to change these drives, or
enter this information in the TINSTALL code. The call with parameters:
               TINSTALL TargetDrv DosDir SourceDrv
MS-DOS Editor  <F1=Help> Press ALT to activate menus          N 00012:001
```

A block of text defined in the MS-DOS Editor appears as inverse text

You can also use specified keys (or combination of keys) to manipulate blocks. If these key combinations seem familiar, it's because they are identical to the key combinations for handling blocks of text using WordStar and the programming language Turbo Pascal.

Marking blocks of text

A block of text must be marked before you work with it. Text can be marked by holding down the Shift key and pressing either →, ←, ↑ or ↓. The MS-DOS Editor displays marked text as highlighted.

Text can also be marked by holding down the left mouse button and dragging the mouse to define the text block.

Only one block of text can be selected at a time. That is, if one passage of text is marked and another passage has already been marked, the first marked passage will no longer be highlighted.

Moving blocks of text

Moving a block of text within a document is a two-step process. You must first move it to the buffer and then move it from the buffer to the new location in the document.

First mark the passage to be moved. Next select the **Cut** command from the **Edit** menu or press Shift + Del. This cuts the passage from the document. Although it disappears from the screen, it has moved to the buffer.

Now position the cursor to the new location within the document. Finally, select the **Paste** command from the **Edit** menu or press Shift + Ins. The passage will reappear in the new location.

Copying blocks of text

Copying a block of text is similar to moving a block.

First mark the passage to be copied. Next select the **Copy** command from the **Edit** menu or press Ctrl + Ins. Then move the cursor to the appropriate location within the document and select the **Paste** command from the **Edit** menu or press Shift + Ins. The copied passage will appear in the new location.

Deleting blocks of text

Deleting a block of text is simple. First mark the passage to be deleted. Then select the **Clear** command from the **Edit** menu or press Del. Text which is deleted is not placed in the buffer and cannot be copied, moved or recovered.

Saving a block of text as a file

Occasionally you'll want to save a block of text as a file. You can do this by using the buffer. First mark the passage and copy it to the buffer using the **Copy** command from the **Edit** menu or Ctrl + Ins (or Shift + Del which cuts the passage into the buffer).

Next select the **New** command from the **File** menu to clear the screen. Then select the **Paste** command from the **Edit** menu or press Shift + Ins to paste the contents of the buffer (containing the desired block of text) to the screen.

Finally, select the **Save As...** command from the **File** menu to give this block of text a new filename and save it onto a diskette or hard drive.

Inserting files to text

Just as you can save a block of text separately from an entire document, you can also insert the contents of a file into a different document. The procedure is similar to the description above.

Use the **Open...** command from the **File** menu to load the file from which you want to cut or copy text. Mark the block of the text (or the entire document). Next, use either Shift + Del, Ctrl + Ins or select the **Cut** command or **Copy** command from the **Edit** menu to place the text block into the buffer.

Select the **New** command from the **File** menu to clear the screen. Then select the **Open...** command of the **File** menu to load the file into which you want to insert the contents of the buffer.

Place the cursor where you want to insert the text and either press Shift + Ins or else select **Paste** from the **Edit** menu to copy the block of text from the buffer into the file.

Cancelling a selection

To cancel a selection (cancel a marked block), simply press any cursor key. The selection will no longer be highlighted.

13.5 Edit Keys

The following table lists the functions of all keys and key combinations.

Simple editor keys

Esc	Aborts commands, closes dialog box.
←	Moves the cursor one character to the left (also Ctrl + S).
→	Moves the cursor one character to the right (also Ctrl + D).
↑	Moves the cursor one line up (also Ctrl + E).
↓	Moves the cursor one line down (also Ctrl + X).
Backspace	Deletes the character to the left of the cursor (also Ctrl + H).
Del	Deletes character in the current cursor position (also Ctrl + G).
Ins	Switches insert mode on or off (also Ctrl + V).
Home	Cursor jumps to the beginning of the line.
End	Cursor jumps to the end of the line.
PgUp	Cursor jumps one screen up (also Ctrl + R).
PgDn	Cursor jumps one screen down (also Ctrl + C).

Extended editor keys

Ctrl + A	Cursor jumps to previous word.
Ctrl + F	Cursor jumps to next word.
Ctrl + N	Rest of the line from current position of the cursor shifts to the next line.
Ctrl + PgUp	Horizontal scrolling page by page (78 characters to the left).
Ctrl + PgDn	Horizontal scrolling page by page (78 characters to the right).
Ctrl + Q S	Cursor jumps to beginning of line.

Ctrl + Q D	Cursor jumps to end of line.
Ctrl + W	Entire screen scrolls one line down. Cursor remains in the same position.
Ctrl + Z	Entire screen scrolls one line up. Cursor remains in the same position.
Ctrl + T	Deletes the next word.
Ctrl + Y	Deletes the line where the cursor is located.
Ctrl + K + ?	Sets a mark; ? stands for a number between 0 and 3. This enables you to mark passages of text while you are working. You can use the Ctrl Q + ?, described below, to move the cursor to these marks. The marks aren't displayed and are not saved when you close the file.
Ctrl + Q + ?	Jumps to mark ?. Ctrl + K + ? must be used to set the marks before using this key combination (see above).

13.6 Moving The Cursor

In this section we'll show you how to move the cursor within the editor. In most cases there are two ways to move the cursor and a way to perform the same action with the mouse. The different keyboard methods are shown below:

Movement	First Method	Second Method
One character right	→	Ctrl + D
One character left	←	Ctrl + S
One line up	↑	Ctrl + E
One line down	↓	Ctrl + X
One word left	Ctrl + ←	Ctrl + A
One word right	Ctrl + →	Ctrl + F
One page up	PgUp	Ctrl + R
One page down	PgDn	Ctrl + C
Beginning of file	Ctrl + Home	
End of file	Ctrl + End	
One window left	Ctrl + PgUp	
One window right	Ctrl + PgDn	
Beginning of line	Home	
End of line	End	

13.7 The Edit Menus

Using the letter keys

Press Alt to activate the menu bar. Then press the highlighted letter key to pull down the individual menus. Pressing the highlighted letter causes the menu to be opened.

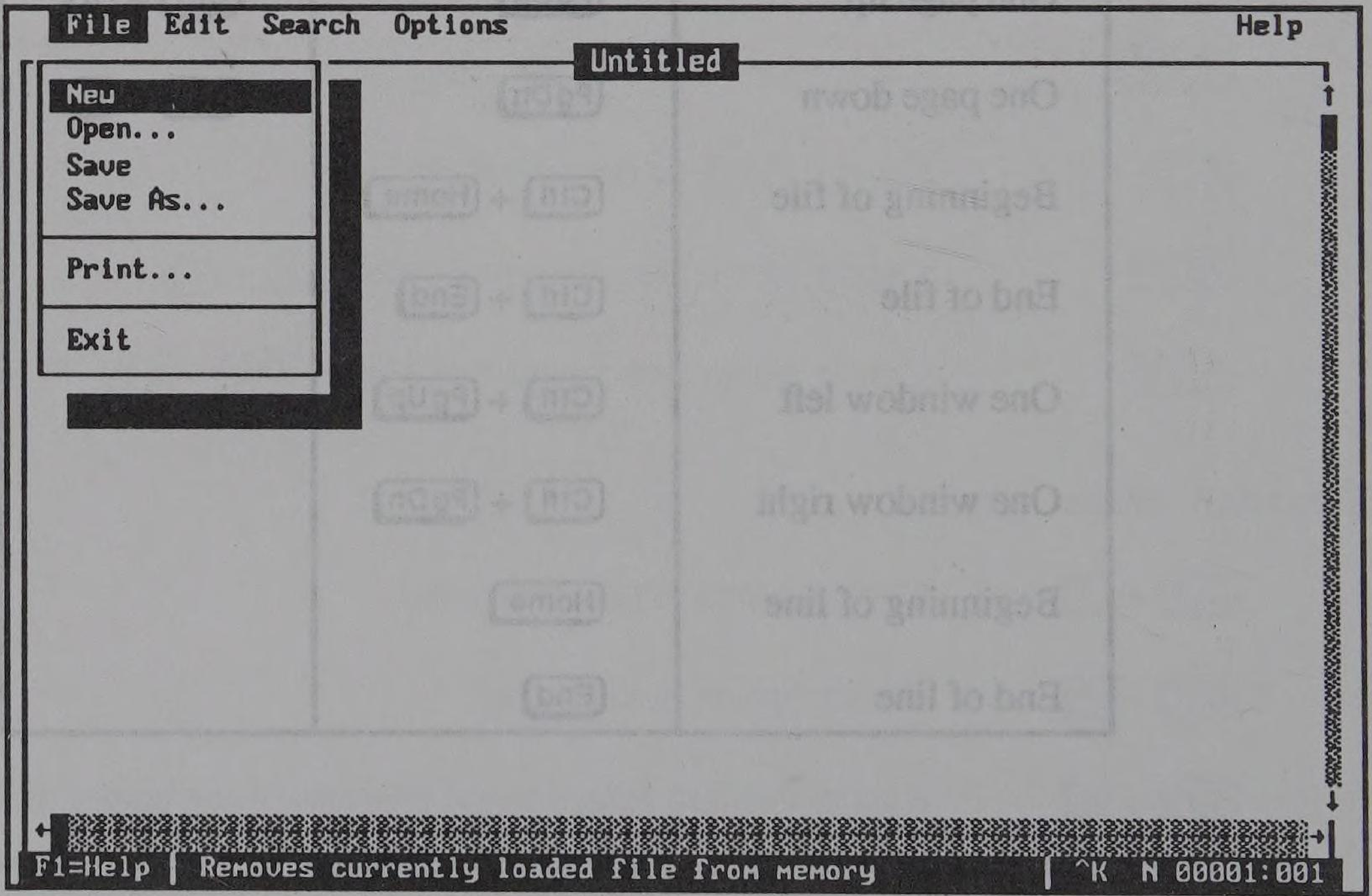
Using the cursor keys

The ↓ and ↑ cursor keys can also be used to open the pull-down menus, while the → and ← cursor keys can be used to select a pull-down menu. Move the area shown in inverse text to the desired command and press Enter to execute the command.

Using the mouse

If you are using the mouse, point and click to the desired menu and the pull-down menu will open. You can also activate individual commands by clicking on them with the mouse.

Now we'll explain how the menus are organized and what the commands do.



```
F1=Help | Removes currently loaded file from memory          | ^K  N 00001:001
```

File menu

The File menu

New Clears the screen and deletes the file contents.

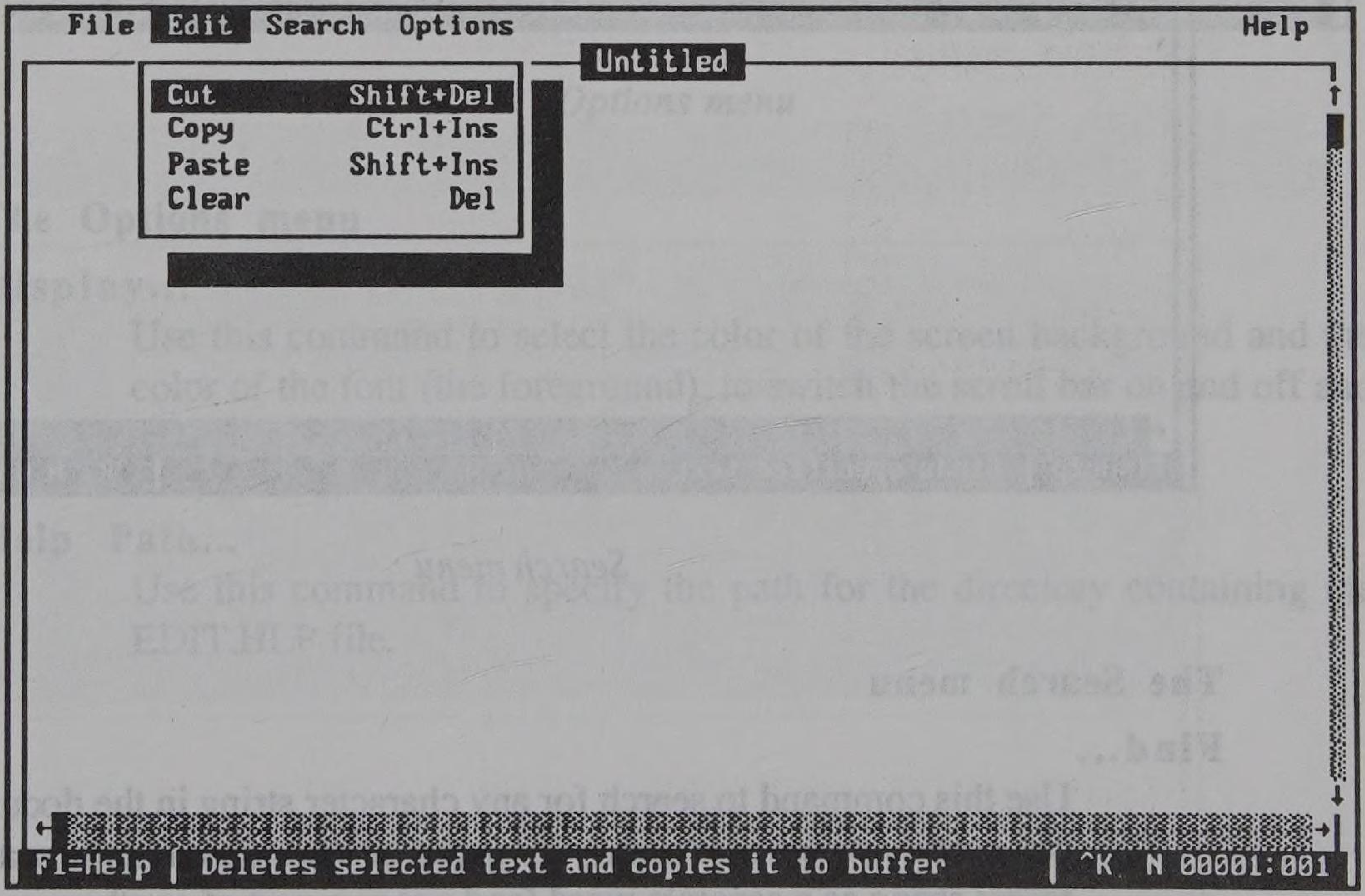
Open...
Use this command to load an existing file from a diskette or hard drive to command memory.

Save Use this command to save a file which has been previously saved. If you haven't saved a file yet, this command prompts you to enter a filename.

Save As...
Use this command to save a new file. You can also use it to save an existing file with a new name.

Print...
Use this command to print either the entire file or a marked block of text on the printer.

Exit Use this command to leave the editor and return to the MS-DOS Shell or command line.


File
Edit
Search
Options
Help
Untitled
Cut Shift+Del
Copy Ctrl+Ins
Paste Shift+Ins
Clear Del

```
F1=Help | Deletes selected text and copies it to buffer      | ^K  N 00001:001
```

Edit menu

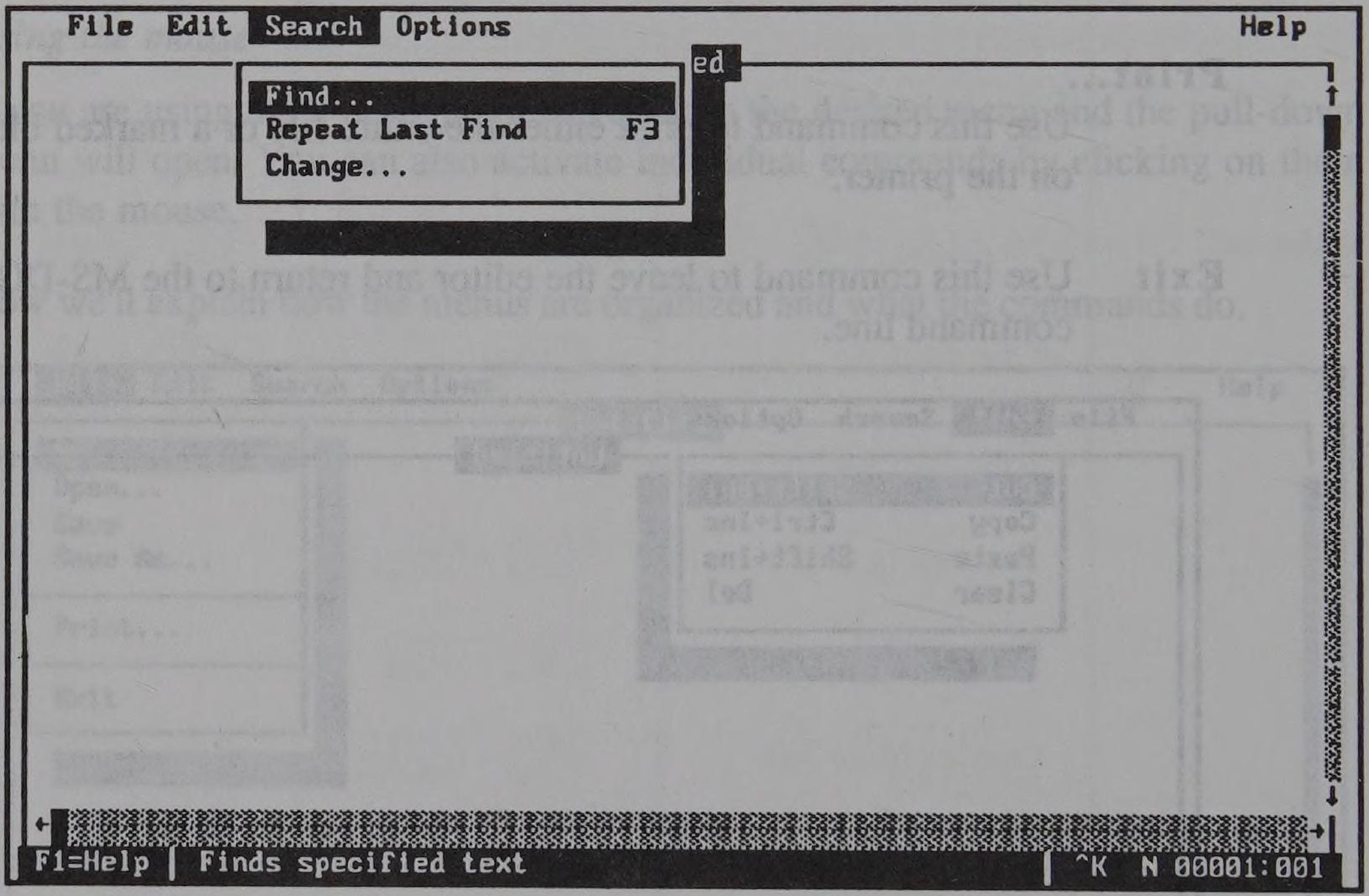
The Edit menu

Cut Use this command to delete a marked block of text from the document and copy it to the buffer.

Copy Use this command to copy a marked block from the document to the buffer.

Paste Use this command to insert the contents of the buffer (block of text) at the position of the cursor in the document.

Clear Use this command to delete a marked block of text. The deleted text is not copied to the buffer.



```
F1=Help | Finds specified text                    | ^K  N 00001:001
```

Search menu

The Search menu

Find...

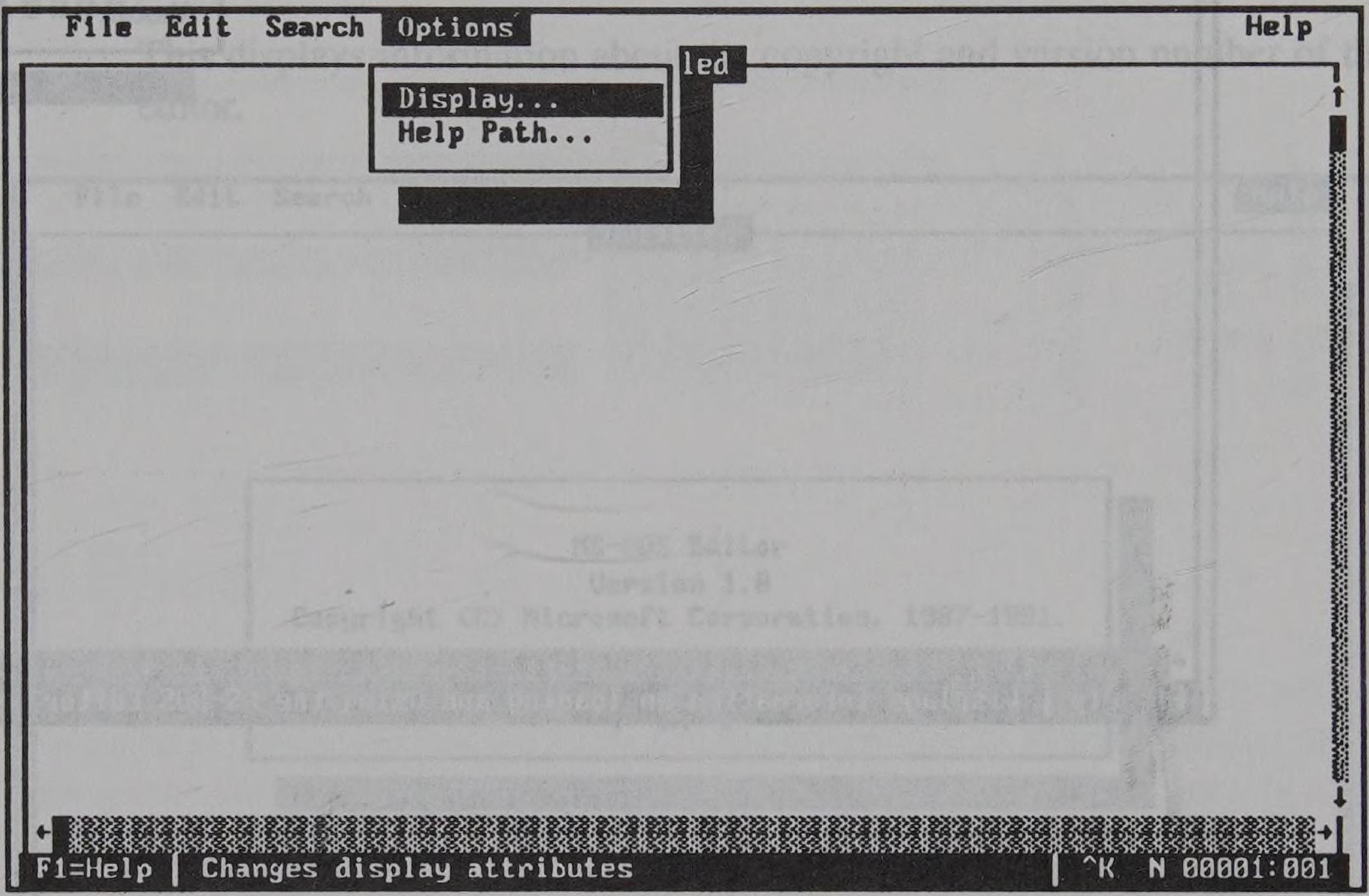
Use this command to search for any character string in the document. You can specify whether to search for upper or lowercase letters, search for the target string as a separate word (and not as a part of another word) as well as the target string itself.

Repeat Last Find

Use this command or press F3 to repeat the last search.

Change...

Use this command to replace any character string in the document with any other string of characters.



```
F1=Help | Changes display attributes                    | ^K  N 00001:001
```

Options menu

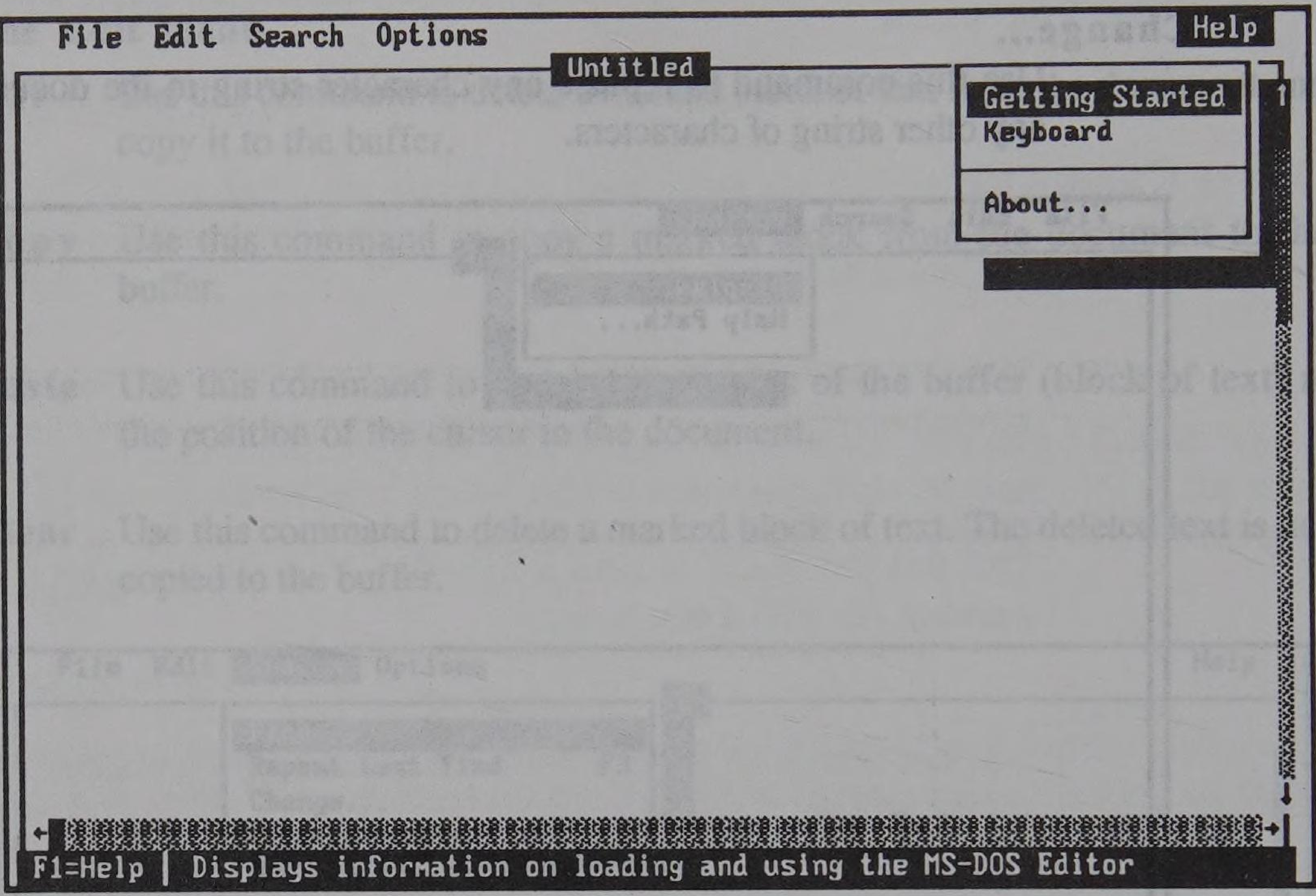
The Options menu

Display...

Use this command to select the color of the screen background and the color of the font (the foreground), to switch the scroll bar on and off and to set the distance between the tabs.

Help Path...

Use this command to specify the path for the directory containing the EDIT.HLP file.


File
Edit
Search
Options
Help
Untitled
Getting Started
Keyboard
About...

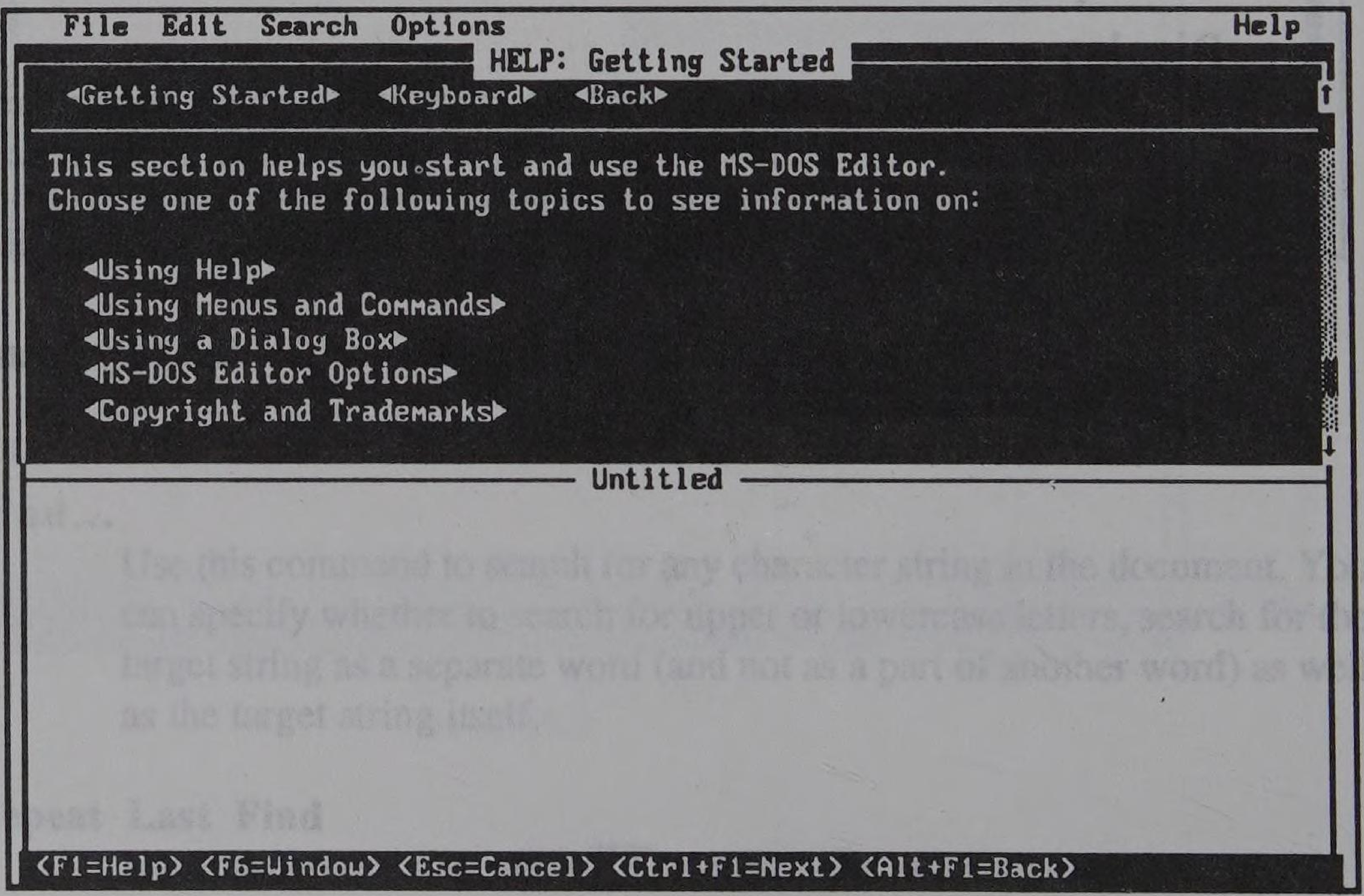
F1=Help | Displays information on loading and using the MS-DOS Editor

Help menu

The Help menu

Getting Started

Displays a help screen that explains how to work with the editor.


File
Edit
Search
Options
Help
HELP: Getting Started
◄Getting Started►
◄Keyboard►
◄Back►
Untitled

This section helps you start and use the MS-DOS Editor.
Choose one of the following topics to see information on:

◄Using Help►
◄Using Menus and Commands►
◄Using a Dialog Box►
◄MS-DOS Editor Options►
◄Copyright and Trademarks►

<F1=Help> <F6=Window> <Esc=Cancel> <Ctrl+F1=Next> <Alt+F1=Back>

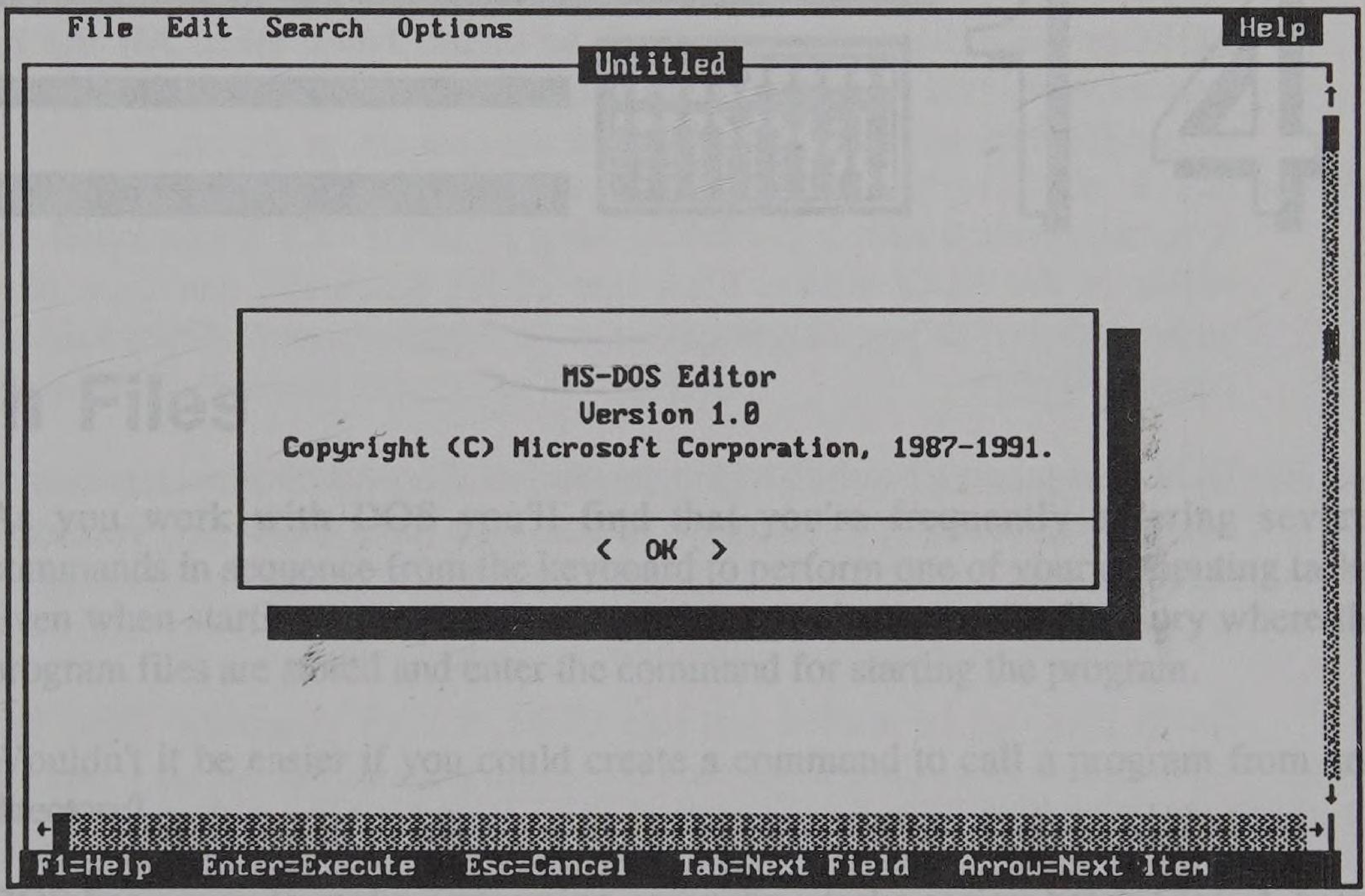
Getting Started help screen

Keyboard

Displays a help screen explaining the editing keys and other special keys.

About...

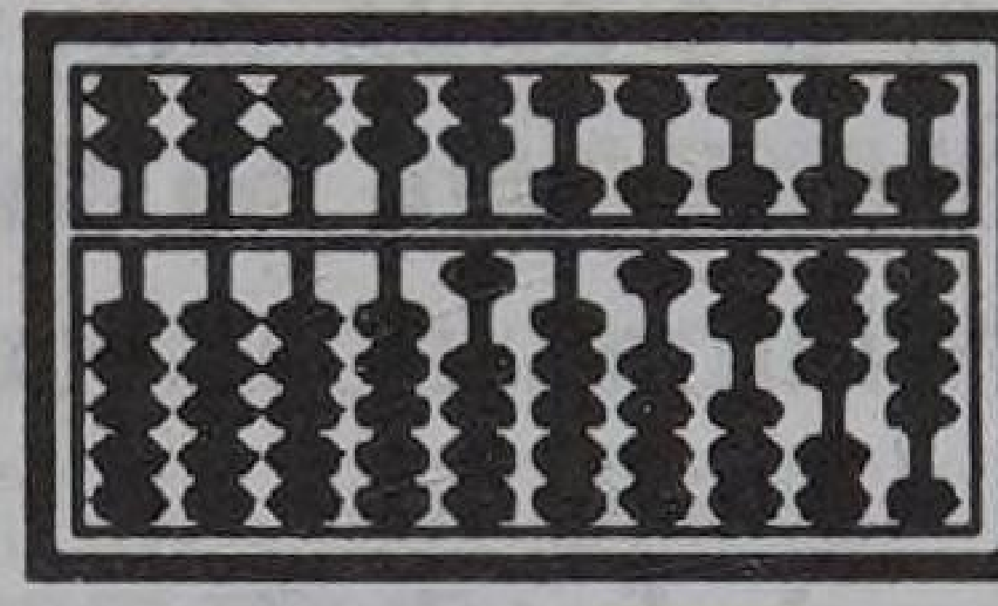
This displays information about the copyright and version number of the editor.


File Edit Search Options
Help
Untitled
< OK >
F1=Help Enter=Execute Esc=Cancel Tab=Next Field Arrow=Next Item

MS-DOS Editor
Version 1.0
Copyright (C) Microsoft Corporation, 1987-1991.

About dialog box

Chapter



14

Batch Files

As you work with DOS you'll find that you're frequently entering several commands in sequence from the keyboard to perform one of your computing tasks. Even when starting a program, you may have to change to the directory where the program files are stored and enter the command for starting the program.

Wouldn't it be easier if you could create a command to call a program from any directory?

DOS has a way to make performing computing tasks easier: batch files. A batch file is an ASCII text file with a .BAT extension. It contains one or more DOS commands, each on its own line. You can think of a batch file as a list of DOS commands.

One batch file that is almost universally used is the AUTOEXEC.BAT file. Starting your computer automatically executes the commands contained in this file.

Batch files can be created with:

- The DOS COPY CON command
- EDLIN
- EDIT
- A word processor, provided the word processor can save text in ASCII format (unformatted text file)

14.1 Batch Files And Programs

Batch files are similar to DOS programs. Programs are files with the extensions .COM or .EXE. They can be called and executed by typing the name of the program or the name of a startup file from the DOS command line.

While special programs, such as assemblers or compilers, are required for creating COM or EXE files, it's quite easy to create batch files. All the batch files presented in this chapter are on the companion diskette for this book. See Appendix E for more information about the companion diskette.

You can create a simple batch file using the DOS COPY command, the EDLIN editor or the EDIT editor. Each line of the batch file can be a normal DOS command, just as you would type it at the system prompt. Each line is ended by pressing Enter.

DOS recognizes a batch file by its extension. Each batch file must have a filename ending with the .BAT extension. A batch file can contain any resident (memory resident) and external (commands which must be loaded before executing) DOS commands and a few special batch commands.

Batch files can be started just like COM or EXE programs. Type its filename without the extension. DOS runs the commands in the file as though they were entered at the system prompt.

When you type a command at the system prompt MS-DOS searches, in a predefined order, for programs and batch files with the same command name (filename).

First, MS-DOS 5.0 searches for a name defined as a DOSKEY macro. If it finds one, the macro is executed. If it doesn't, DOS checks its list of resident (internal) commands for the same name and executes it if found. Otherwise, DOS assumes that it's an external command or batch file.

The search continues in the current directory. In the order listed below, DOS looks for a file with the same name but having one of these extensions:

- .COM
- .EXE
- .BAT

If it can't find a filename with one of these extensions in the current directory, DOS then searches each directory defined by the search PATH.

So if you type a command, similar to the following, at the system prompt:

```
TEST Enter
```

1. DOS searches for a macro named TEST
2. DOS searches for an internal command named TEST
3. DOS searches the current directory for a file named TEST.COM
4. DOS searches the current directory for a file named TEST.EXE
5. DOS searches the current directory for a file named TEST.BAT
6. DOS searches each directory in the search PATH for a file named TEST.COM, TEST.EXE or TEST.BAT

If a batch file has the same name as an existing DOS command, the DOS command is executed rather than the batch file.

You can overcome this limitation by specifying the file extension along with the command. For example, if you created a batch file called FORMAT.BAT in the DOS directory and want to call this file instead of the DOS FORMAT.COM command, type:

```
FORMAT.BAT
```

This only works with external DOS commands. It can't be used to override an internal DOS command, such as DIR.

Example

If a batch file is named DIR.BAT and you type DIR, DOS will not run the batch file. Rather, it will always run the internal command, DIR.

If a batch file named FORMAT.BAT is stored in the directory of DOS commands and you enter FORMAT, DOS won't run the batch file. It will execute the DOS command FORMAT.COM instead.

However, you can replace all of the external DOS commands with batch files. Just save the batch files in their own directory (C:\BAT) and specify this directory in the search path ahead of the DOS directory:

```
PATH C:\BAT;C:\DOS
```

You can create a special version of the FORMAT command as a batch file that automatically displays the contents of a diskette before formatting it. This batch file might look like the following batch file called FORMAT1.BAT:

To avoid any confusion with the FORMAT command this batch file is named FORMAT1.BAT on the companion diskette.

```
@ECHO OFF
IF "%1" == "" GOTO error
DIR %1
ECHO Press any key to format disk
ECHO Press CTRL-C to abort
PAUSE
C:\DOS\FORMAT %1 %2 %3 %4
GOTO End
:Error
ECHO You must specify a drive.
:End
ECHO ON
```

Remember if this batch file was named FORMAT.BAT, the batch file directory must be before the DOS directory in the PATH or DOS will find the FORMAT command and not the FORMAT.BAT batch file. The line "C:\DOS\FORMAT %1 %2 %3 %4" must specify the correct path for the DOS FORMAT command. Otherwise the batch file would call itself and not the "real" FORMAT command.

If you run this batch file, and the directory displayed contains files that you don't want to lose, the process can be ended easily when the following message appears:

```
Press any key to continue. . . .
```

The parameter "%1" represents the first parameter passed to the batch file, which is usually the drive label. The parameters "%2" through "%4" represent any other parameters that were specified. The batch file passes these parameters to the DOS FORMAT command. Several requirements must be satisfied in order for this process to run as intended:

- There can't be a macro, defined by DOSKEY, named FORMAT1. If a macro has been defined, type FORMAT1.BAT to run the batch file.
- If the batch file was named FORMAT.BAT, it must be in a directory that has been specified ahead of the DOS directory in the search path.
- If the batch file was named FORMAT.BAT, the current directory cannot be the directory containing the DOS commands. DOS would find the real FORMAT command instead of starting the batch file through the search path. This can be overcome by entering the filename extension as part of the command, FORMAT.BAT.

Be very careful when naming batch files the same name as DOS commands, they may directly affect the operation of other batch files. To avoid any confusion with the FORMAT command the batch file presented in this section is named FORMAT1.BAT on the companion diskette.

14.2 Creating And Running Batch Files

COPY CON, EDLIN, EDIT or a word processor can be used to create batch files.

We recommend that you store batch files in their own directory instead of in the DOS directory on a hard drive. If you want to have the batch files available to run at all times, use PATH to set a search path to the directory containing the batch files and enter this directory name before the DOS directory name.

Using COPY CON to create batch files

To create a batch file in the current directory, type the command line shown below:

```
COPY CON FILENAME.BAT
```

Enter each of the command lines at the keyboard. Then press Enter after each line. A line can contain up to 127 characters. To complete the process and save the file, enter Ctrl + Z and press Enter.

➡ If a file with the specified name already exists, it is overwritten by the new file without any warning. Before using COPY CON to create a file, make sure that the target directory doesn't contain a file that has the same name as the file you are creating.

The COPY CON command is useful for creating small batch files, but there are better ways to create longer batch files. For example, you can only correct the current line when you enter the batch file.

Also, COPY CON cannot be used to change a batch file. If you must change a batch file with COPY CON, the entire batch file must be re-entered.

Using EDLIN to create batch files

It's more convenient to use EDLIN to create a batch file. Although EDLIN is far from the best editor, it still lets you change batch files easily.

Here's a simple example of how to create a batch file with EDLIN.

Just as with COPY CON, carefully consider the filename for a batch file. In this example, we'll create a batch file that first changes the current directory to the DOC directory and then starts the Microsoft Word word processor from this directory.

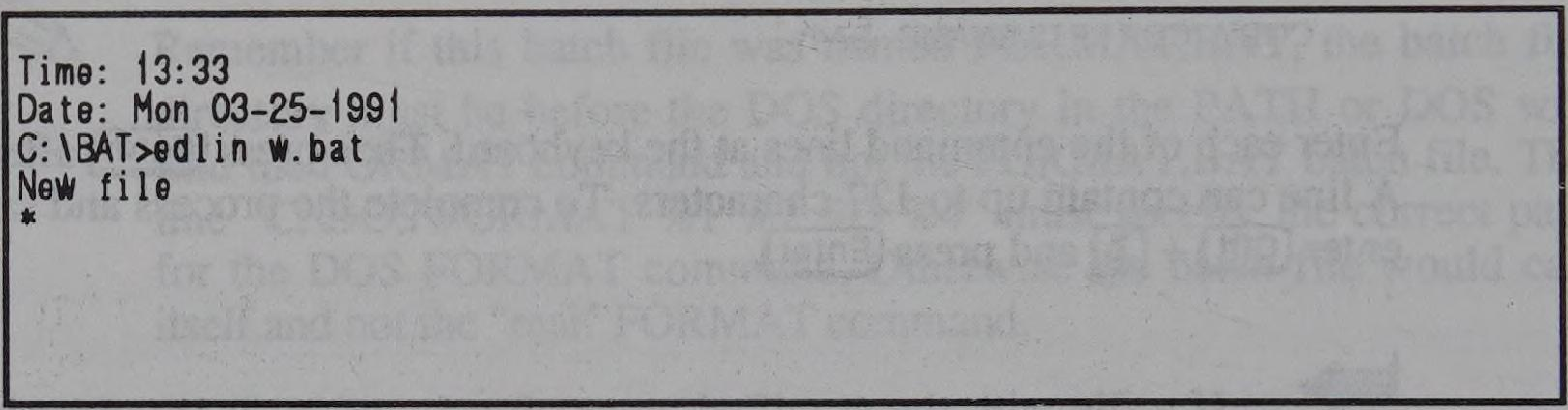
We'll call the batch file W.BAT. This will enable you to enter a short command (the W and Enter keys) to start Word.

Start EDLIN with the filename of the file to be created, W and the .BAT file extension, as follows:

```
EDLIN W.BAT
```

Remember that EDLIN, unlike COPY CON, is a transient DOS program. This means that to start the program, you must either select the directory containing the EDLIN command or the path to the proper directory must have been specified by PATH.

You'll see a message that a new file is being created and you will see an asterisk (*) in the next line. This asterisk indicates that you are in the command mode of EDLIN.



```
Time: 13:33
Date: Mon 03-25-1991
C:\BAT>edlin w.bat
New file
*
```

Entering EDLIN

You can't enter any text in this mode. You can only activate one of the commands of EDLIN. To enter input mode, press I for INSERT and then press Enter.

A new line appears with a 1, colon and an asterisk. You're now in input mode and can type the DOS commands that the batch file will later run. It is important that each DOS command be in its own line.

Press Enter to complete each line and move the cursor to the next line. Let's assume that you keep your document files in a directory named C:\DOC, and that you keep WORD.EXE in the C:\WORD directory. The lines are numbered consecutively. In this example, you would have to type the following commands on the screen:

```
CD C:\DOC Enter
C:\WORD\WORD Enter
```

These are the commands that the batch file will execute.

To save this command sequence as a batch file, first return to command mode by pressing Ctrl + Z or F6 when the cursor is located at the start of a new line. Then press Enter.

The asterisk reappears on the far left side of the screen. This indicates that you're in command mode. Enter E for EXIT and press Enter. The batch file is saved and you are returned to the system prompt of the DOS command line.

Copy the batch file to a directory that is in the search PATH. The batch file can then be started from any directory.

See Chapter 12 for more information on using EDLIN.

Using EDIT to create batch files

EDIT is probably the best way to create a batch file.

To create a batch file in the current directory with EDIT, start the editor as follows:

```
EDIT FILENAME.BAT Enter
```

Enter EDIT and the name of the batch file to be created. Remember to include the file extension .BAT.

After pressing Enter, the editor screen appears and you can begin typing the lines of the batch file. DOS commands for the batch file must be on separate lines. Press Enter at the end of each line to start a new line.

After entering all of the lines, save the batch file. Press Alt + F to select the **File** menu, then press S to select **Save**. The file is automatically saved under the name you already specified.

You must exit the editor before you can use the batch file. Press Alt + F to return to the **File** menu, then press X to select the **Exit** command. You will return to the system prompt on the DOS command line.

See Chapter 13 for more information on using EDIT.

Using a word processor to create batch files

Any word processor that can save files as unformatted ASCII files can also be used to create batch files.

14.3 Using Batch Files

There are various ways to start batch files. A batch file can even be started by another batch file. This section explains the requirements for starting a batch file and provides important information about calling a batch file.

Starting a batch file

Similar to COM and EXE files, a batch file is started by entering its name. The system prompt reappears after the last command in the batch file is executed.

When calling a batch file, remember the following points:

Do not define a macro, using DOSKEY, with the same name as an existing batch file. If a macro has been defined by DOSKEY, you'll have to enter both the filename and the .BAT extension to start the batch file.

- The batch file must be in the current directory or in a directory which is defined in the search path specified by PATH.
- There can't be any COM or EXE files, in the current directory, with the same name. If COM or EXE files with the same name exist, they will be started instead of the batch file.

Requirements for running a batch file

If you start a batch file from a diskette and remove the diskette while the batch file is running, DOS displays a message similar to:

```
Insert disk with batch file
Press any key to continue. . .
```

Replace the diskette with the batch file and press any key.

Running a batch file from within a batch file

Normally if you start a batch file which starts a second batch file, DOS will terminate all batch files and return you to the system prompt of the command line when the last command in the second batch file is completed.

To be able to run a second batch file in the same way you would call a subroutine from BASIC and then continue with the next command in the original batch file, use CALL:

```
CALL FILENAME.BAT
```

DOS executes the called batch file and then runs the next command in the file following the CALL.

Aborting a batch file

A batch file can be stopped at any time by pressing Ctrl + C. The computer will display the prompt:

```
Terminate batch job (Y/N)?
```

Enter Y to abort the batch file. This prompt will not be displayed if the computer is formatting a diskette.

If it takes a long time for DOS to respond to Ctrl + C, use the BREAK ON option in your CONFIG.SYS file. For more information on using BREAK, refer to Chapter 18.

14.4 Displaying And Modifying Batch Files

It's very easy to display, print and modify batch files.

Displaying a batch file

Use the TYPE command to display a batch file on the screen:

```
TYPE FILENAME.BAT
```

If the batch file is longer than one screen page, use Ctrl + S to stop the output. Press any key to continue.

You can also use the following to display a large batch file one page at a time:

```
TYPE FILENAME.BAT | MORE
```

Printing batch files

Use the following to print a batch file to a printer:

```
TYPE FILENAME.BAT >LPT1
```

The greater than character (>) redirects the output to a parallel printer connected to LPT1. If your printer is connected to the first serial port, use COM1 instead of LPT1.

Printing several batch files

You can print several batch files, either directly or in the background. In the second case, you can enter more commands from the system prompt while the batch files are being printed.

Printing several batch files directly

Use the COPY command to print several batch files if the output is redirected to the printer. Use wildcards for the filenames to be printed. If your printer is not connected to the first parallel port, use the proper port identifier in place of LPT1, as shown below. For example, to print all of the batch files in the current directory, use the following:

```
COPY *.BAT LPT1
```

The filename and the contents of all of the batch files are sent to the printer.

Printing several batch files in the background

You can use the PRINT command to print more than one batch file. Using this method lets you continue working while the files are being printed. To do this, enter the filenames to be printed after the PRINT command, separated by spaces:

```
PRINT FILE1.BAT FILE2.BAT FILE3.BAT
```

You can also use wildcards when specifying the filenames. To print all of the files, in the current directory, with the .BAT extension, use the following command line:

```
PRINT *.BAT
```

If several files are selected to be printed, PRINT stores the information in a special list, called the print queue. PRINT enters the specified files in this list and prints them one after the other. There are also special commands for removing files from the print queue.

Before using PRINT for the first time, the port must be set. It's best to do this from the AUTOEXEC.BAT file with the following line:

```
PRINT /D:LPT1
```

As a rule, the print queue can contain no more than ten files. If a wildcard is used, and the filenames apply to more than ten files, the excess files won't be printed. The /Q switch specified with PRINT can be used to increase the maximum number of files the print queue can hold.

More information about PRINT can be found in Chapter 18.

Editing a batch file

Either EDLIN or EDIT can be used to edit batch files.

Using EDLIN to change batch files

Start EDLIN along with the name of the file:

```
EDLIN FILENAME.BAT
```

After the file is loaded, the screen displays the message "End of input file". As we previously explained, enter commands after the asterisk (*).

Use the L (List) command to display the first 23 lines of a batch file. In order to start the display at a specific line, enter the starting line number in front of L.

To change a line in a batch file, enter the number of the line. The screen will display the line. You can then edit the line. After pressing Enter, EDLIN accepts the changes to the line.

If you need to insert new lines into a batch file beginning at a specific line, enter the number of the line that follows the new lines and the I (Insert) command. Now you can insert as many new lines as needed. To end insertion, move the cursor to the beginning of a blank line and press Ctrl + Z and press Enter.

The D (Delete) command removes lines from a batch file. Enter the number of the line to be deleted in front of the command. To delete more than one line, enter the number of the first line to be deleted followed by a comma, then enter the number of the last line to be deleted followed by D. Such a command would look like:

```
x,y D
```

"x" is the number of the first line to be deleted and "y" is the number of the last line to be deleted.

Enter E (Exit) to save the changes and leave EDLIN. The file is saved and you return to the command line of DOS at the system prompt.

To exit EDLIN without saving any changes, enter a Q and reply to the displayed prompt with Y.

Using the MS-DOS Editor to modify a batch file

Modifying batch files with the MS-DOS Editor (EDIT) is probably the easiest method you'll find. Start EDIT with the name of the batch file to be changed as a parameter.

```
EDIT FILENAME
```

The EDIT screen will appear, displaying the contents of the batch file. Use the cursor keys to move the cursor to the text to be changed.

To insert characters into the batch file, simply type them at the desired location.

Use DEL to remove characters beneath the cursor.

To insert a new line in a batch file, place the cursor at the end of the line immediately before the point the new line should be inserted and press Enter. A new blank line is inserted. Type the new commands in the new line.

To save a batch file after making changes, press Alt + F to select the **File** menu and then press S for **Save**. The file is automatically saved under its original filename.

To save the modified batch file under a different name, press Alt + F to select the **File** menu and then press A for **Save As**. Enter a new name for the file in the dialog box. After pressing Enter, the file is saved under the new name.

You must exit from the MS-DOS Editor to test a modified file. Press Alt + F for the **File** menu and then press X for **Exit**.

14.5 Batch File Commands

All DOS commands can be used in batch files. In addition, there are a few special commands that are used exclusively for batch file programming.

Displaying a message

The ECHO command has two different jobs.

Messages can be displayed using ECHO:

```
ECHO Message
```

Here, "message" can contain any text, including spaces. However, there are some characters with special meanings that are not allowed.

If you need to display a blank line use the special invisible character. Press Alt + 2 5 5 to create the invisible character. Remember to place a space between ECHO and this character. ECHO will also display a blank line if a period is entered immediately after the command:

```
ECHO.
```

Setting the screen to display all command lines

ECHO can be used to determine whether or not DOS displays the command lines of the batch file before executing the commands.

```
ECHO ON
```

This command causes all command lines that follow it to be displayed on the screen. To switch off displaying all command lines, type:

```
ECHO OFF
```

Typing ECHO without any parameters displays the current status.

Suppressing the screen display of a single command line

You can also suppress the display of a single command when ECHO is switched on. Place the “@” character in front of the command line.

```
@COMMAND LINE
```

Even if you use ECHO OFF to switch off the display of the command lines, the first command line still appears on the screen. Use "@" to prevent the first command line from being displayed. Place the "@" character at the beginning of the first line:

```
@ECHO OFF
REM The rest of the batch file follows
```

Example

Here's an example of how ECHO works.

We want to create a batch file that makes the directory containing text files (we'll call this directory DOC) the current directory and display the directory page by page. This batch file is called TEXTDIR.BAT. It is on the companion diskette in the BAT directory.

Here's what the batch file looks like:

```
CD \DOC
DIR /P
```

This batch file displays its command lines on the screen. To prevent this display, insert the following lines in the batch file:

```
@ECHO OFF
CD \DOC
DIR /P
ECHO ON
```

These lines prevent any lines following ECHO OFF from being displayed on the screen.

Placing "@" in front of ECHO OFF prevents this line from being displayed on the screen as well.

At the end of a batch file, always remember to set ECHO back to ON. In some situations the system prompt might not appear on the screen after a batch file. This could be confusing.

The following example demonstrates a different way to use ECHO to display a screen message.

Make the following changes to the batch file we just created:

```
@ECHO OFF
ECHO Selects the DOC directory and displays the directory
CD \DOC
DIR /P
ECHO ON
```

When this batch file, called TEXTDIR2.BAT, is run you will see the following line displayed on the screen:

```
Selects the DOC directory and displays the contents
```

ECHO can be used to display relevant information as a batch file is run. This information becomes even more important as you create larger batch files. ECHO can be used to display messages explaining what is happening in a large batch file.

Comments in batch files

The REM command can be used to insert comments into a batch file. DOS does not execute these lines. REM commands can be used by the batch file creator to document the way in which the batch file will operate. A comment line always has the following form:

```
REM Comment
```

Comment lines are especially useful in long batch files; they give you a general idea of the operation of the batch file. You can use comment lines to mark subroutines, etc. This helps you find your place when revising the batch file.

Here's an example of using the REM command.

We want to create a batch file that always displays the current directory sorted according to filenames. We'll call this batch file DIRSORT, and will include a comment line so that when we revise it later, we will immediately know what the batch file does.

```
@ECHO OFF
REM ╔═════════════════════════════════════════╗
REM ║ Batch for displaying a sorted directory ║
REM ╚═════════════════════════════════════════╝
DIR /O:N
ECHO ON
```

The following characters can be used to make a frame around comments:

```
Alt+201 ╔ = Alt+205 ╦ Alt+203        ╗ Alt+187

Alt+186 ║                            ║ Alt+186

Alt+204 ╠          ╬ Alt+206         ╣ Alt+185

Alt+186 ║                            ║ Alt+186

Alt+200 ╚ = Alt+205 ╩ Alt+202        ╝ Alt+188
```

When this batch file is called, the comment lines won't be displayed. In fact, unless you use the TYPE command to display the batch file on the screen, you won't see them again until you revise the batch file.

Testing conditions in batch files

The IF command lets you check for different conditions as a batch file is run. If a condition is satisfied, the rest of the line following the condition is executed. Otherwise, the execution skips to the next line.

Comparing character sequences

The following line can be used to compare two character strings:

```
IF CHARACTER_STRING1==CHARACTER_STRING2 COMMAND
```

If both character strings are identical, the command is executed; otherwise the command is skipped and the next command line is executed.

The IF comparisons are not limited to character strings. Parameters and environment variables that have been passed to the batch file can also be compared.

Most of the time you'll be comparing a parameter or an environment variable within the batch file, such as:

```
IF %COMSPEC%==C:\COMMAND.COM ECHO OK
```

When making such a comparison, the parameter or the environment variable may not exist. In this example, if the variable COMSPEC doesn't exist, the following line would be displayed:

```
IF ==C:\COMMAND.COM ECHO OK
```

This line generates an error message because there is nothing to compare on the left side of the equal sign. A simple solution to this problem consists of adding a character on both sides so that the correct syntax is assured:

```
IF %COMSPEC%!==C:\COMMAND.COM! ECHO OK
```

If the COMSPEC variable contains the desired contents, then a correct comparison is still ensured:

```
IF C:\COMMAND.COM!==C:\COMMAND.COM! ECHO OK
```

However, if the variable is not present, at least DOS finds something to compare on both sides of the equals sign; so it doesn't generate a syntax error message:

```
IF !==C:\COMMAND.COM! ECHO OK
```

Another common method consists of adding two quotation marks on both sides of the comparison:

```
IF "%COMSPEC%"=="C:\COMMAND.COM" ECHO OK
```

This also prevents a syntax error when a variable isn't present.

DOS differentiates between upper and lowercase letters when comparing text strings.

Checking whether a file exists

Use the key word EXIST in a batch file to check whether a file exists:

```
IF EXIST FILE COMMAND
```

In the preceding command line, if the file exists, the batch file executes the command; otherwise it will skip to the next command line.

Example

To determine whether the system file AUTOEXEC.BAT exists in the root directory of the hard drive and make a backup copy of it, use the following command line:

```
IF EXIST C:\AUTOEXEC.BAT COPY C:\AUTOEXEC.BAT C:\AUTOEXEC.BAK
```

You can also use EXIST to check whether directories or drives exist.

To check whether a directory exists, use:

```
IF EXIST \NAME\NUL COMMAND
```

To check whether a specific directory exists before displaying the directory, type:

```
IF EXIST \TEMP\NUL DIR \TEMP
```

To determine whether a drive exists, use:

```
IF EXIST DRIVE_LETTER:\NUL Command
```

To check whether a drive exists before attempting to copy to it, type:

```
IF EXIST A:\NUL COPY AUTOEXEC.BAT A:
```

Checking the return value of programs

Upon completion, a program can supply a return value. This value is ordinarily used to signal an error that occurs while the program is running.

Many DOS commands supply a return value other than 0 when an error occurs. This variable, ERRORLEVEL, can be used to check the return value from within a batch file. ERRORLEVEL 0 means that an error was not found.

The syntax for a command to check the ERRORLEVEL is:

```
IF ERRORLEVEL NUMBER COMMAND
```

If the return value is greater than or equal to a specified number, the command is executed; otherwise it skips to the next command line.

The following is a more detailed explanation of what ERRORLEVEL means and what it does.

The meaning and purpose of ERRORLEVEL

When a command is entered after the system prompt and a mistake occurs when the command is being executed, an error message is displayed. At that point you can correct the command or eliminate the cause of the error.

But what can you do when there are a series of command lines within a batch file that automatically run? Checking ERRORLEVEL is only a partial solution.

What is an ERRORLEVEL?

An ERRORLEVEL is the return value - a report on the success or failure of a program. We'll explain later in this chapter how a program supplies such a value. For now you need to know that each program started by DOS, and even each DOS command, can report a return value between 0 and 255.

Usually programs are started by the command interpreter, which receives the return value, but doesn't display it. On the other hand, if you call the program from within a batch file, the command interpreter notes the returned value and saves it in the variable ERRORLEVEL.

Asking for an ERRORLEVEL

You can determine an ERRORLEVEL only from within a batch file. Many DOS commands supply a return value other than 0 if an error occurs. If an ERRORLEVEL of 0 is returned, it means no error has occurred.

As we mentioned earlier, the syntax of a line that you can use to determine the ERRORLEVEL in a batch file looks like this:

```
IF ERRORLEVEL NUMBER COMMAND
```

If the return value is greater than or equal to a specified number, the command is executed; otherwise it skips to the next command line. Unfortunately you cannot check for a specific value, but only for a minimum value. This makes it difficult to be precise when responding to the error level.

Example

The following batch file, called &FORMAT.BAT, formats a diskette in drive A:. If an error occurs during this process, or if you abort the process by pressing Ctrl + C, the batch file displays a message.

```
@ECHO OFF
FORMAT A:
IF ERRORLEVEL 1 GOTO error
GOTO end
:error
ECHO.
ECHO Formatting not successful!!!
:End
ECHO ON
```

In this example, after the batch file calls FORMAT, it checks if there is an ERRORLEVEL greater than 1 (in other words, whether an error has occurred). The batch file doesn't examine the return value more closely.

What information is provided by ERRORLEVELs

ERRORLEVEL was made a part of DOS for error checking. Three basic conditions apply:

- ERRORLEVEL 0 means: no error found.
- ERRORLEVEL > 0 means: error found.
- There is no set meaning for a special value. Instead this depends on the DOS command. However, the higher the return value, the more serious the error.

ERRORLEVEL with BACKUP

To fully understand the meanings of the different return values, you must examine the commands more closely. For example, the DOS BACKUP command could return the following values:

0	No error, data backup successful.
1	No files found for backup.
3	Aborted by user (Ctrl + C or abort after errors in accessing file).
4	Wrong parameters (wrong path for source directory or an invalid option specified when calling the command).

As you can see, the seriousness of the error increases with the value of the number returned. So ERRORLEVEL 1 is more of a warning than a "real" error.

In order to respond to an error condition and take appropriate action within a batch file, check for the highest possible value first.

The following batch file, called FORMATA.BAT, is a simple example of error checking following a FORMAT command:

```
ECHO OFF
FORMAT A:
IF ERRORLEVEL 4 GOTO error4
IF ERRORLEVEL 3 GOTO error3
ECHO No error, formatting successful
GOTO End
:error4
ECHO Illegal drive or parameter. Possible problems with diskette.
GOTO End
:error3
ECHO Formatting terminated by CTRL + C
GOTO End
:End
ECHO ON
```

You should check the ERRORLEVEL whenever possible when calling a program from a batch file.

Also, check for possible errors immediately after starting the program. ERRORLEVEL may be changed by other DOS commands. DOS only saves one ERRORLEVEL at a time. As soon as a new command changes the return value, the previous value is lost.

ERRORLEVEL in DOS commands

Not all DOS commands return error values and each command uses only certain values. Also, different versions of DOS use different return values. You may need to experiment a little.

The following table shows the ERRORLEVEL of some DOS commands along with short explanations. This is only a partial list.

ATTRIB

1	wrong parameter or file not found
3	abort

BACKUP

1	no files found for backup
3	terminated by user
4	wrong parameter or error found

DISKCOMP

1	invalid drive, wrong syntax

2	terminated by user

DISKCOPY

1	invalid drive, wrong syntax
2	abort

FASTOPEN

1	wrong parameter

FIND

2	wrong parameter

FORMAT

3	abort
4	error found (wrong name specified for diskette, hard drive)

JOIN

1	invalid parameter

LABEL

1	drive not available/invalid

MODE

1	wrong parameter

RESTORE

1	no files found
3	aborted by user
4	source contains no backup data, invalid parameters

SUBST

1	invalid parameter

XCOPY

2	abort
4	invalid number of parameters, invalid path

How do you determine the ERRORLEVEL for commands

It is not easy to determine all of the possible ERRORLEVELs for a command because you have to produce the error first and then ask for the ERRORLEVEL.

Although some DOS manuals do have information about the return values for the different errors, the information isn't always complete.

The following batch file, ERROR.BAT, will help you determine as many of the return values as possible (up to the value of 10 for a DOS command).

```
ECHO OFF
%1 %2 %3 %4 %5 %6 %7 %8 %9
ECHO.
IF ERRORLEVEL 10 ECHO 10
IF ERRORLEVEL 9 ECHO 9
IF ERRORLEVEL 8 ECHO 8
IF ERRORLEVEL 7 ECHO 7
IF ERRORLEVEL 6 ECHO 6
IF ERRORLEVEL 5 ECHO 5
IF ERRORLEVEL 4 ECHO 4
IF ERRORLEVEL 3 ECHO 3
IF ERRORLEVEL 2 ECHO 2
IF ERRORLEVEL 1 ECHO 1
ECHO ON
```

Now, to try a command, just enter ERROR in front of the command line. With this method, you can test the different conditions and get the return value for each one.

For example, to determine which errors the DOS BACKUP command returns when you use the wrong syntax to call it, use the following command line:

```
ERROR BACKUP A B
```

As a result, the following is displayed:

```
No target drive specified
ECHO is off
4
3
2
1
```

So we see that when no target drive is specified, the BACKUP command returns an ERRORLEVEL of 4.

First, the normal error message for BACKUP is displayed, then the ERRORLEVEL. Because we intentionally kept the batch file simple, the lowest values are also displayed.

Reversing the condition

The keyword NOT can be used to reverse the three possible conditions.

```
IF NOT CHARACTER_STRING1==CHARACTER_STRING2 COMMAND
```

If the character strings are not equal, command will be executed; otherwise it skips to the next command line.

✎ If you use parameters or environment variables within such a comparison, use quotation marks on both sides to avoid syntax errors if the variables are not displayed.

```
IF NOT EXIST File Command
```

If the file is not present, command is executed; otherwise it skips to the next command line.

```
IF NOT ERRORLEVEL NUMBER COMMAND
```

If the return value is not greater than or equal to the specified number, the command is executed; otherwise the next command line is executed.

Example

```
ECHO OFF
IF NOT EXIST C:\AUTOEXEC.BAT ECHO System file not present!
REM Rest of the batch file
```

It's a good idea to use GOTO (see below) if you want to execute several commands in response to one condition being met.

Jumping over command lines

Usually, the individual command lines in a batch file are executed one after the other. Sometimes it's necessary to jump over command lines and continue execution at a different place in the batch file. This is done by defining a label and using GOTO to have DOS continue at the label.

The label must be at the beginning of a line and must start with a colon. The label can be up to eight characters long. If a longer label is used, DOS ignores all characters after the eighth one. A typical GOTO command would look like the following:

```
GOTO Label
:
:
:
:Label
```

When DOS encounters a GOTO command, execution branches to the specified line with the label and continues at the command in the following line. The rest of the line with the label is ignored. It doesn't matter whether you define the label before or after GOTO. In any case, DOS branches to the specified label even if it precedes the GOTO command within the batch file.

Don't use any special characters as part of the name of a label. These characters could cause problems with running the command correctly. Spaces are permitted.

Example

Frequently, GOTO is used in a conditional command. The following batch file, called &GOTO.BAT, displays a message if no parameters are specified when it is called:

```
ECHO OFF
IF "%1"=="" GOTO Syntax
ECHO All clear
GOTO End
:Syntax
ECHO You must specify a parameter!
:End
ECHO ON
```

The first line of this batch file checks whether a parameter was specified. If no parameter was specified and if "%1" is not present (it's equal to ""), the rest of the line is executed. Then the batch jumps directly to the label "Syntax" and displays the message "You must specify a parameter!". If a parameter was specified, the batch skips the rest of the line (GOTO Syntax is not executed) and continues with the next line.

Using GOTO and IF provides you with the option of asking questions and having other commands depend on the responses to the questions.

Looping in batch files

The FOR command can be used in loops to have parts of a batch file execute several times.

```
FOR %%V in (list) DO COMMAND %%V
```

%%V	This variable "V" (placeholder) accepts all of the values from the list, one after the other.
(list)	Specifies the list of files, to which you want the command to apply. You can list each file individually (File1 File2 File3) or by using a wildcard (*.TXT). In the first case, the variable "V" contains File1 File2 File3, in the second case it contains all available files with the .TXT extension.

COMMAND

After DO, specify the command to be performed for each value in List. Only one command can be used.

You can also use the parameters of the batch file "%0" to "%9" (see below) or environment variables.

Like many other batch commands, FOR can be used outside of batch files, directly from the system prompt. When FOR is used from the system prompt, only one % sign is used with the variable.

The FOR command cannot be nested. This means that you cannot have another loop with FOR within a loop.

To run more than one DOS command within a loop, use CALL to call a new batch file.

Example

The following batch file displays all files with the .BAT extension on the screen:

```
FOR %%V IN (*.BAT) DO TYPE %%V
```

To run this line from the system prompt, type:

```
FOR %V IN (*.BAT) DO TYPE %V
```

Only the .BAT files in the current directory are displayed.

The following batch file searches all files with the .DOC extension, in the current directory, for the character sequence "DOS".

```
FOR %%V IN (*.DOC) DO FIND "DOS" %%V
```

The following example displays the names of all COM or EXE files in the current directory:

```
FOR %%V IN (*.COM *.EXE) DO ECHO %%V
```

Calling batch files from batch files

As mentioned above, CALL can be used to run other batch files as subroutines:

```
CALL FILE
```

If DOS encounters this command, it starts the batch file specified in the command line. After the called batch file is complete, DOS continues execution with the command line after CALL. Parameters can be passed to the called batch file:

```
CALL FILE PARAMETER
```

Example

Suppose you have a batch file, named TYPEM.BAT, that displays a specified file on the screen page by page:

```
TYPE %1 | MORE
```

For "%1" DOS uses the first parameter specified. Usually this is a specified filename.

To have this batch file display the contents of C:\AUTOEXEC.BAT file, type the following command after the system prompt:

```
TYPEM C:\AUTOEXEC.BAT
```

You can also specify the parameter C:\AUTOEXEC.BAT when calling another batch file from within a batch file:

```
CALL TYPEM C:\AUTOEXEC.BAT
```

You can start a batch file, such as TYPEM.BAT, without using CALL by specifying the name. In this case the batch file would not be executed as a subroutine, and the entire batch file process would end when the secondary batch file (such as TYPEM.BAT) was completed.

The following is another practical use for CALL.

You already know how to apply one command to several variables using FOR...DO. When using FOR...DO you can only use a single command within the loop.

This is where CALL is useful. For example, to use more than one command on several files, use the FOR...DO loop and CALL after DO to call a new batch file containing the commands you want to use on the files. This will cause the commands to be executed.

```
FOR %%a IN (AUTOEXEC.BAT CONFIG.SYS) DO TYPE %%a
```

This line in a batch file displays the AUTOEXEC.BAT and CONFIG.SYS files on the screen. To copy the files onto a diskette as well, the batch needs to execute a few more program lines. However, by using CALL, you can call a new batch file that does this for you. Change the first batch file to:

```
FOR %%a IN (AUTOEXEC.BAT CONFIG.SYS) DO CALL Command %%a
```

The COMMAND.BAT file would include the lines:

```
TYPE %1
COPY %1 A:
```

PAUSE

PAUSE lets you interrupt the execution of a batch file. The batch will stop and wait until a key is pressed. PAUSE recognizes Ctrl + C to terminate the batch file. Using PAUSE provides new possibilities for batch programming.

When the PAUSE command is used, the following message is displayed:

```
Press any key to continue . . .
```

After pressing a key, DOS continues executing the batch file with the next command line.

Text can be entered after the PAUSE command, but DOS won't display it on the screen as it does with ECHO.

Example

The following is an example of a batch file that deletes all files with the .BAK extension. PAUSE allows you to decide whether the files should be deleted. Using one of the methods we previously discussed, create the following batch file called DELBAK.BAT:

```
@ECHO OFF
ECHO All files in the current directory with
ECHO the .BAK extension will be deleted.
ECHO To cancel, press CTRL+C
PAUSE
DEL *.BAK
ECHO ON
```

When this batch file is started, the screen display indicates what's going to happen (the delete operation) and how to cancel it (press Ctrl + C). If you press the Ctrl + C key combination, the batch file terminates and you are returned to the DOS system prompt. The files with the .BAK extension won't be deleted in this case.

To delete all of the files with .BAK extensions, press any key and the batch file will continue running, executing the following lines. Since this example actually deletes files, use it carefully. It's best to try it out first on a practice diskette, especially if you are the least uncomfortable with using the file.

SHIFT

Parameters can be passed to a batch file. Parameters are passed in special variables "%1" through "%9". You can use SHIFT to change the contents of these replaceable variables:

```
SHIFT
```

DOS shifts the existing parameters one place to the left. When SHIFT is used, "%1" receives the contents of "%2", "%2" receives the contents of "%3", etc. If you specified a tenth parameter, it is now in "%9". The contents of "%0" are lost. Before the first Shift, "%0" contains the name of the batch file (see below).

Example

The following batch file, &GCOPY.BAT (group copy) copies all of the files or file groups that were specified onto a diskette in drive A:. To do this, all of the specified files are shifted to the parameter "%1" by the SHIFT command. The batch file ends when "%1" no longer contains any files:

```
@ECHO OFF
:Start
IF "%1"=="" GOTO EndItAll
COPY %1 A:
SHIFT
GOTO Start
:EndItAll
ECHO ON
```

To copy all files with the .TXT and .BAK extensions, use the following command line to call the batch file:

```
&GCOPY *.TXT *.BAK
```

When first called, "%1" contains the TXT parameter and "%2" contains the BAK parameter. After the SHIFT is encountered the first time, "%2" is empty and "%1" contains .BAK. When the batch file executes SHIFT the second time, "%1" is empty and the batch files ends by jumping to the label EndItAll.

14.6 Batch Files And Parameters

One special advantage of batch files is that you can design them rather generally and then enter additional information as parameters when you call them.

When writing a batch file program, variables are indicated as a percentage sign followed by a number between 1 and 9. This means that nine parameters can be specified.

%0	Contains the name of the batch file as specified in the command line that calls the batch.
%1	Contains the first parameter.
%2	Contains the second parameter.
.	
.	Contains the third through eighth parameter.
.	
%9	Contains the ninth parameter.

You can also access additional parameters by using SHIFT to shift the parameter list (see previous section).

The parameters "%0" to "%9" are replaced by the contents of each variable before the command line is executed. This will let you check for correct usage by displaying the information when ECHO is switched on (with ECHO ON).

To check whether a parameter has been specified, use IF to check the variable contents:

```
IF "%1"=="" COMMAND
```

If nothing has been stored in "%1" then the parameter is unused and command is executed. You can use GOTO to branch to the end of the batch file.

DOS always places values in the variables in sequence. So if the variable "%2" is empty, then "%3" and all subsequent variables are also empty.

When writing a batch file that uses parameters, make sure that you specify the exact number of parameters before the batch file is started. If there aren't enough parameters specified, the batch file usually cannot perform its task correctly. Specifying too many parameters can also cause problems.

The following example, DELD.BAT, demonstrates how to determine the proper number of parameters. The batch file deletes a directory that has been passed as a parameter, but first checks that exactly one parameter was specified. Otherwise the batch displays the proper syntax on the screen:

```
@ECHO OFF
REM DELD.BAT deletes the contents of a directory
REM and removes the directory.
REM The directory cannot contain any subdirectories.
IF "%1"=="" GOTO Syntax
IF NOT "%2"=="" GOTO Syntax
ECHO Directory %1 being deleted.
IF EXIST %1\*.* DEL %1\*.*
RD %1
GOTO EndItAll
:Syntax
ECHO Syntax Error!
ECHO DELD.BAT removes a directory
ECHO Call: DELD Directory
ECHO Example DELD DATA
:EndItAll
ECHO ON
```

First the batch file checks if a parameter exists. If no parameter has been specified, the batch process ends and displays the correct syntax.

If the first parameter (%1) has been specified, a check is made for more than one parameter. If more than one parameter has been specified, the batch file again branches to display the proper syntax.

The batch file will display "Directory %1 being deleted" if the proper syntax was entered. DOS replaces "%1" with the specified parameter.

Before the batch uses RD to remove the directory, it must be certain that there aren't any more files in the directory. The batch file uses the IF EXIST statement for this check. If any files are present, DEL is used to delete them. Because of the wildcards used in the specification (*.*), DEL automatically displays a security prompt.

After all files have been removed, RD can remove the directory. If the specified directory doesn't exist, RD displays an error message.

Solving batch file problems involving parameters

Sometimes batch file parameters can be troublesome.

The parameters "%0" to "%9" are replaced by the contents of the parameters specified in the command line before it is executed. DOS replaces "%1" with the contents of the first specified parameter. If ECHO is switched on, you can check the command lines for the replaced variables as they are displayed on the screen.

Displaying programs

The following batch file, SHOWPRG.BAT, demonstrates the flexibility of the options for using parameters in batch files. It shows all of the executable programs for a specified filename. The filename can be specified with either wildcards or with the complete filename in the sequence that DOS executes them (.COM, .EXE or .BAT).

```
@ECHO OFF
ECHO %0 shows all specified programs
FOR %%a in (%1.com, %1.exe, %1.bat) DO ECHO %%a
@ECHO ON
```

To find out whether there is more than one FORMAT command with different extensions in the current directory, type:

```
SHOWPRG FORMAT
```

Path specifications and wildcards are allowed. All DOS commands beginning with the letter "F" can be displayed by typing the following command line:

```
SHOWPRG C:\DOS\F*
```

This line should display the following:

```
C:\DOS\FDISK.EXE
C:\DOS\FORMAT.COM
C:\DOS\FASTOPEN.EXE
C:\DOS\FIND.EXE
```

14.7 Batch Files And Environment Variables

In addition to the variables "%0" to "%9", environment variables can be examined or used in batch files.

Using environment variables

To use an environment variable, enclose the variable name in percentage signs:

```
%Variable%
```

DOS replaces this with the current contents of the environment variable before executing the command line. If the environment variable has not been defined, nothing is inserted.

One method for checking whether an environment variable has been defined is to use IF to check whether the variable has any contents:

```
IF NOT "%Variable%" == "" Command
```

If the environment variable has not been defined, the batch file executes the command which follows in the command line. For example, the command could display an error message or jump to the end of the batch file.

Example: Extending a path

The following batch file, &ADDPATH.BAT, adds another directory to the existing search path. The DOS PATH command cannot be used to do this. All directories for the search path must be specified at one time in the command line:

```
@ECHO OFF
IF "%1"== "" GOTO EndItAll
PATH %PATH%;%1
:EndItAll
PATH
@ECHO ON
```

For example, to add C:\WORD to the existing search path, call the batch file with:

```
&ADDPATH C:\WORD
```

The batch adds the new directory (WORD) to the existing path specification and then displays the new search paths.

Defining an environment variable

The following command can be used to define environment variables in batch files:

```
SET Variable=Contents
```

Instead of environment variables, specify the name of the variable; enter the desired contents for the variable after the equals sign. You can even use spaces for the contents of the variable.

Use SET without entering any parameters to display all defined environment variables.

Example

Suppose that you want to place the name of a user in a variable. Simply type the following after the system prompt:

```
SET User=Meyer
```

You are not limited to defining environment variables from batch files. They can also be defined from the system prompt. However, you can only use the contents of environment variables within batch files.

For example, to store the name of the batch file that is currently being executed in a variable, use the following line in the batch file:

```
SET Batchname=%0
```

To display the name of the last batch file that was executed, use the following command line:

```
SET | FIND "Batchname"
```

First SET determines all defined variables, then FIND looks for the line with the desired variable name. Remember that FIND is case sensitive. This means that CASE differentiates between upper and lowercase letters. Variable names are stored by DOS as uppercase letters.

14.8 Forbidden Characters In Batch Files

There are some characters that you either cannot use or that are not easy to use. These characters are:

%	DOS uses this character to identify the parameters "%1" to "%9". To use this character in text and display it with ECHO, it must be entered twice (%%).
<	You use the less than character "<" to redirect the input from the keyboard to a new device or file. You cannot use this character in batch files.
>	This character redirects the output to a device other than the standard output (the screen) and cannot be used in batch files.
|	You use this character (pipe symbol) to pass the output of a program to a different program. This character is also forbidden for other uses in batch files.

If you want to display the characters "<", ">" or "|" from a batch file, write the screen output to a separate file and use the following command to copy it onto the screen:

```
COPY FILE CON
```

14.9 DOSKEY And Batch Files

You can use the DOSKEY utility to create macros, which in many ways are identical to simple batch files. DOSKEY can also be used to build batch files almost automatically. We'll discuss both of these uses for DOSKEY.

Using DOSKEY to create batch files

DOSKEY can be used to create batch files almost automatically.

First DOSKEY must be installed into memory. If you do not already have it installed, type:

```
DOSKEY Enter
```

From this point on, DOSKEY saves all of the command lines entered (up to the limits of the history buffer). These command lines can later be executed again. To display all of the lines in the history buffer, type:

```
DOSKEY /HISTORY
```

An even easier way to display all of the previous commands is to press F7. Any of the commands in the history buffer are listed, just as they would appear in a batch file.

You can use this list to create a batch file with DOSKEY.

To create a batch file based on these previously entered command lines, use the following procedure:

1. Press Alt + F7 to clear all of the commands in the history buffer.
2. Type the DOS command lines at the keyboard the way you ordinarily do. DOSKEY stores these commands in the history buffer as it executes each command for you.
3. To create a batch file, called WORK.BAT, from the history buffer, type the following command line:

   ```
   DOSKEY /HISTORY >WORK.BAT
   ```

This redirects the contents of the history buffer to a file named WORK.BAT. This *almost* creates your batch file. DOSKEY also records the last command line entered (the command line for creating the batch file: "DOSKEY /HISTORY >WORK.BAT") and writes it to the batch file. Use an editor to remove this line. You can either use EDLIN or EDIT:

```
EDIT WORK.BAT
```

Delete the last line and save the completed batch file.

Sometimes you will enter a series of commands and later realize that you could have created a batch file from these lines to simplify your work. Since you didn't use Alt + F7 to clear the history buffer before you started entering commands, other unwanted commands are also stored there. You can still write the history buffer to a file:

```
DOSKEY /HISTORY >FILE.BAT
```

Start EDIT, delete the unwanted command lines and save the file again. Don't forget to remove the last line which created FILE.BAT.

Using DOSKEY to create macros - "mini batch files"

Besides making it very convenient to enter, edit and execute commands at the system prompt, the DOSKEY utility also lets you define *macros*. A macro is an abbreviation for a series of DOS commands. Instead of typing each command at the keyboard, you type the abbreviation and DOS performs each of the commands defined by the macro. In effect, a macro is similar to a batch file. However, a macro cannot contain all of the special batch file commands which make batch file programming so powerful.

To define a macro, use the following syntax:

```
DOSKEY MACRONAME=COMMAND_LIST
```

As an example, let's define a macro that displays the names and paths, in alphabetical order, of all the BAT files in the subdirectories of the current drive. Furthermore, we want to display them without the directory header and summary format. Here's how we might define the macro:

```
DOSKEY DIRBAT=DIR *.BAT /p/s/on/b Enter
```

After you define this macro, you can list all of the batch files in the subdirectories of the current drive by simply typing:

```
DIRBAT Enter
```

DOSKEY lets you combine commands in a macro. To do this, separate the commands by the characters "$T". So, to change the above example to list all BAT files in all subdirectories of the current drive you would type:

```
DOSKEY DIRBAT=cd \$T dir *.bat /p/s/on/b Enter
```

In this case, the DIRBAT macro first changes to the root directory before listing the BAT files.

"Automatic" DOSKEY macros

As you become familiar with macros, you'll find that they're indispensable for performing many of your computing tasks. So it's important to have an easy way to use predefined macros.

There are two basic ways to simplify macro definition:

1. Insert the DOSKEY macro definitions into your AUTOEXEC.BAT file.
2. Run a batch file, from your AUTOEXEC.BAT, which contains the DOSKEY macro definitions.

You'll first have to define each of your macros and then test them to ensure that they work properly. When you've defined them to DOSKEY, you can then list them by typing:

```
DOSKEY /MACROS Enter
```

By redirecting the above command, you can write these definitions to a file on your hard drive. For example, to create a file called MACDEFS.BAT, you would type:

```
DOSKEY /MACROS >MACDEFS.BAT Enter
```

All of the definitions are then written to the designated file. To turn the file into a true batch file, use EDIT or EDLIN to add the DOSKEY command to each of the lines in the MACDEFS.BAT file. The MACDEFS.BAT file is now ready to run as a stand alone batch file to define each of the DOSKEY macros.

Delete the first line, and save the completed batch file.

Sometimes you will enter a series of commands that [illegible] you could [illegible] commands, [illegible] are also stored there [illegible] history [illegible]

```
DOSKEY /HISTORY > [illegible]
```

Start EDIT, delete the unneeded command lines and save the file again. Don't [illegible]

Using DOSKEY to create macros - "mini batch files"

[illegible] commands at the system prompt, the DOSKEY utility also lets you define macros. A macro is an abbreviation for a series of DOS commands. Instead of typing each command at the keyboard, you type the abbreviation and Enter [illegible] commands defined by the macro. In effect, a macro is similar to a batch file. However, a [illegible] batch file.

To define a macro, use the following syntax:

```
DOSKEY MACRO_NAME=COMMAND_LIST
```

[illegible] paths, in [illegible] drive. [illegible] summary format. Here's how we might define the macro:

```
DOSKEY DIRBAT=DIR *.BAT /p/s/on/b Enter
```

After you define this macro, you can list all of the batch files in the subdirectories of the current drive by simply typing:

```
DIRBAT Enter
```

DOSKEY lets you combine commands in a macro. To do this, separate the commands by the characters "$T". So, to change the above example to list all BAT files in all subdirectories of the current drive you would type:

```
DOSKEY DIRBAT=cd \ $T dir *.bat /p/s/on/b Enter
```

Chapter 15

The MS-DOS Shell

The Shell is an alternate way of working with DOS. Instead of having to type commands at the system prompt, the Shell lets you select the desired commands and programs from pull-down menus.

SAA Interface

The Shell was first introduced in DOS in Version 4.0. For MS-DOS 5.0, the appearance and operation of the Shell has changed to conform to the SAA (System Application Architecture) standard.

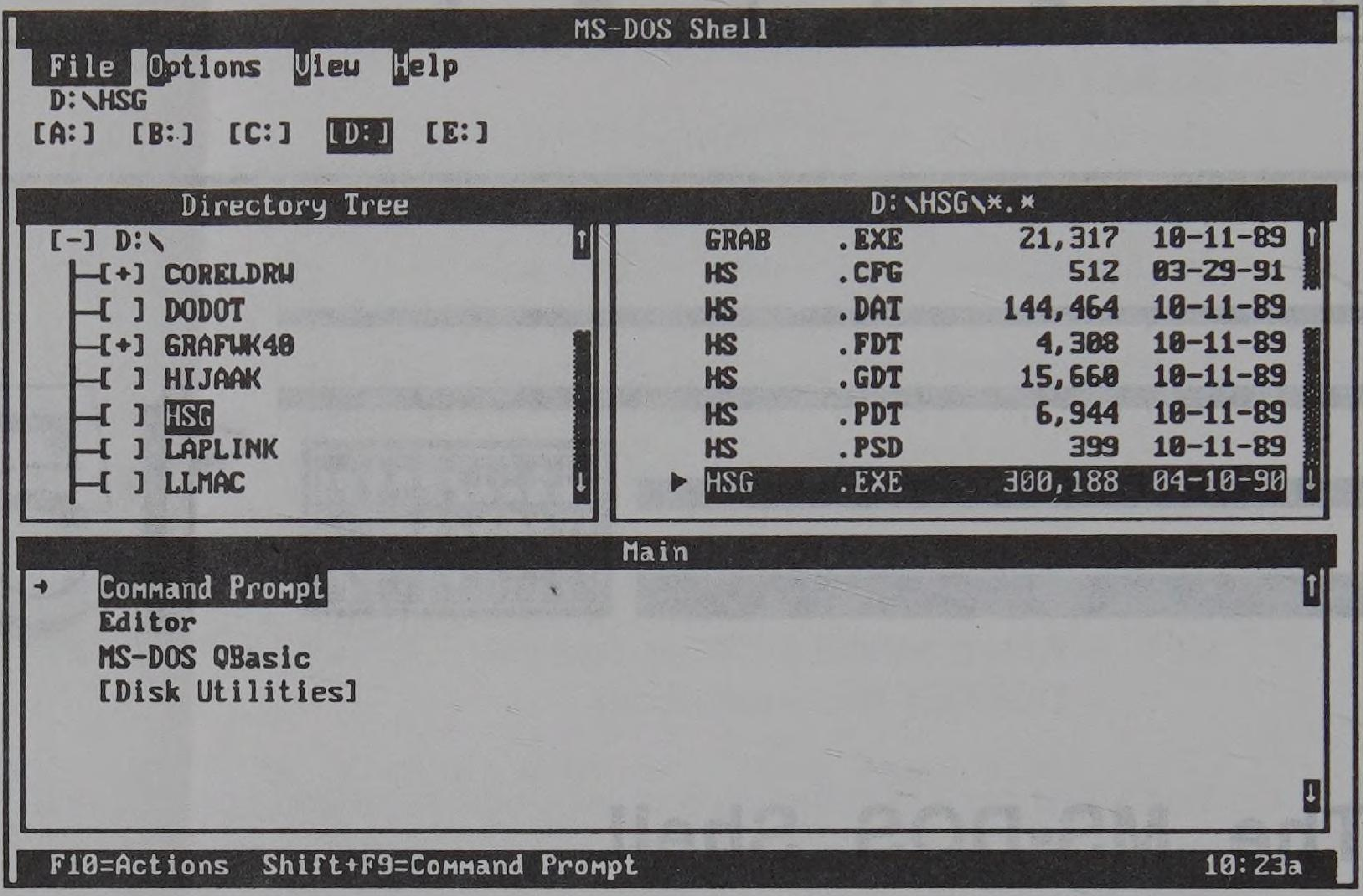
The SAA standard was designed to create a user interface that is both uniform and user-friendly. This can significantly decrease the time needed to learn how to use different programs. The SAA standard also contains guidelines for exchanging data, which makes it easier to share data between different programs.

Simple operation with the mouse

The Shell lets you use the most important operating system commands and applications by using an easy-to-operate menu system. It's especially easy to use the Shell if you have a mouse attached to your computer. This makes it even easier to move, select and use the various commands. You'll find that using the mouse simplifies your work with DOS.

Customizing the MS-DOS Shell

The Shell is very versatile. You can use the Shell as it comes from Microsoft or you can adapt it to fit your own needs by integrating new groups and programs.



```
                                MS-DOS Shell
 File  Options  View  Help
  D:\HSG
 [A:]  [B:]  [C:]  [D:]  [E:]

          Directory Tree                              D:\HSG\*.*
  [-] D:\                                 GRAB     .EXE       21,317  10-11-89
   ├─[+] CORELDRW                         HS       .CFG          512  03-29-91
   ├─[ ] DODOT                            HS       .DAT      144,464  10-11-89
   ├─[+] GRAFWK40                         HS       .FDT        4,308  10-11-89
   ├─[ ] HIJAAK                           HS       .GDT       15,660  10-11-89
   ├─[ ] HSG                              HS       .PDT        6,944  10-11-89
   ├─[ ] LAPLINK                          HS       .PSD          399  10-11-89
   ├─[ ] LLMAC                          ▶ HSG      .EXE      300,188  04-10-90

                                        Main
 →     Command Prompt
       Editor
       MS-DOS QBasic
       [Disk Utilities]

 F10=Actions  Shift+F9=Command Prompt                                   10:23a
```

The MS-DOS Shell

The Shell is designed for two very different types of users. It helps novice users operate a computer and use DOS without completely understanding complicated commands and the command interpreter.

It also lets experienced users set up a computer that is custom-made to their individual needs. The Shell has many options for customizing the user interface that exceed the capabilities of other user interfaces.

15.1 Advantages Of The MS-DOS Shell

The MS-DOS Shell gives the user a predefined way to "talk" to MS-DOS. Since the Shell graphically presents file, directory and program information on the screen, it can organize and simplify the routine computing tasks that the user needs to perform.

How does the Shell simplify a user's work? Let's use a short example - making a copy of a file. To perform this task from the command line you might type:

```
COPY myfile.txt newfile.txt
```

Of course this isn't a very difficult task to perform. But in order to type the above command to perform this simple task, we had to know several things:

- The name and spelling of the command - Copy
- That the name of the source file must be typed after the command
- The exact spelling of the source file - myfile.txt
- That the name of the target file must be typed after the source filename
- The exact spelling of the target file - newfile.txt
- That a space must be inserted between the source and target filename

However, by using the Shell you can easily select the file from a list and select the task from a pull-down menu. This is easier because you don't have to memorize the command name and syntax.

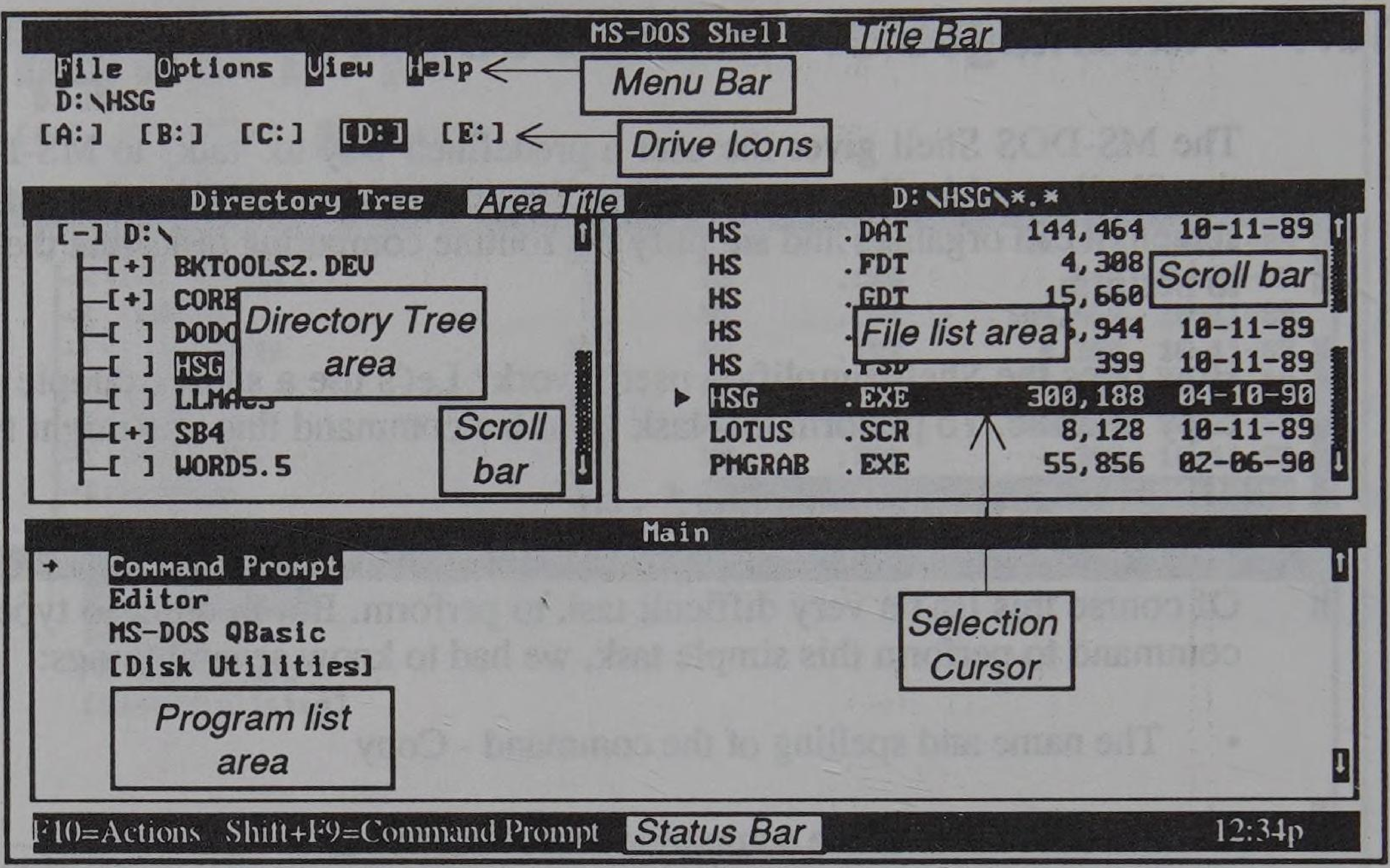
Installing the MS-DOS Shell

When you install MS-DOS 5.0 on your hard drive, you copy four files that are required by the Shell. These files are: DOSSHELL.EXE, which is a program file; DOSSHELL.HLP, which is an information file to teach you about using the Shell; DOSSHELL.VID, which contains hardware information for the Shell; DOSSHELL.INI, which contains configuration data for the Shell. You won't be able to run the Shell unless all of these files were installed during installation.

15.1.1 MS-DOS Shell Basics

The MS-DOS Shell simplifies two important areas of personal computing:

- Managing files and directories
- Running programs


MS-DOS Shell
Title Bar
File Options View Help
Menu Bar
D:\HSG
[A:] [B:] [C:] [D:] [E:]
Drive Icons
Directory Tree
Area Title
[-] D:\
[+] BKTOOLS2.DEV
[+] CORE
[ ] DODO
[ ] HSG
[ ] LLMA
[+] SB4
[ ] WORDS.5
Directory Tree area
Scroll bar
D:\HSG\*.*
HS .DAT 144,464 10-11-89
HS .FDT 4,308
HS .GDT 15,660
HS ,944 10-11-89
HS 399 10-11-89
HSG .EXE 300,188 04-10-90
LOTUS .SCR 8,128 10-11-89
PMGRAB .EXE 55,856 02-06-90
Scroll bar
File list area
Main
Command Prompt
Editor
MS-DOS QBasic
[Disk Utilities]
Program list area
Selection Cursor
F10=Actions Shift+F9=Command Prompt
Status Bar
12:34p

The basic areas of the MS-DOS Shell

Managing files and directories with the MS-DOS Shell

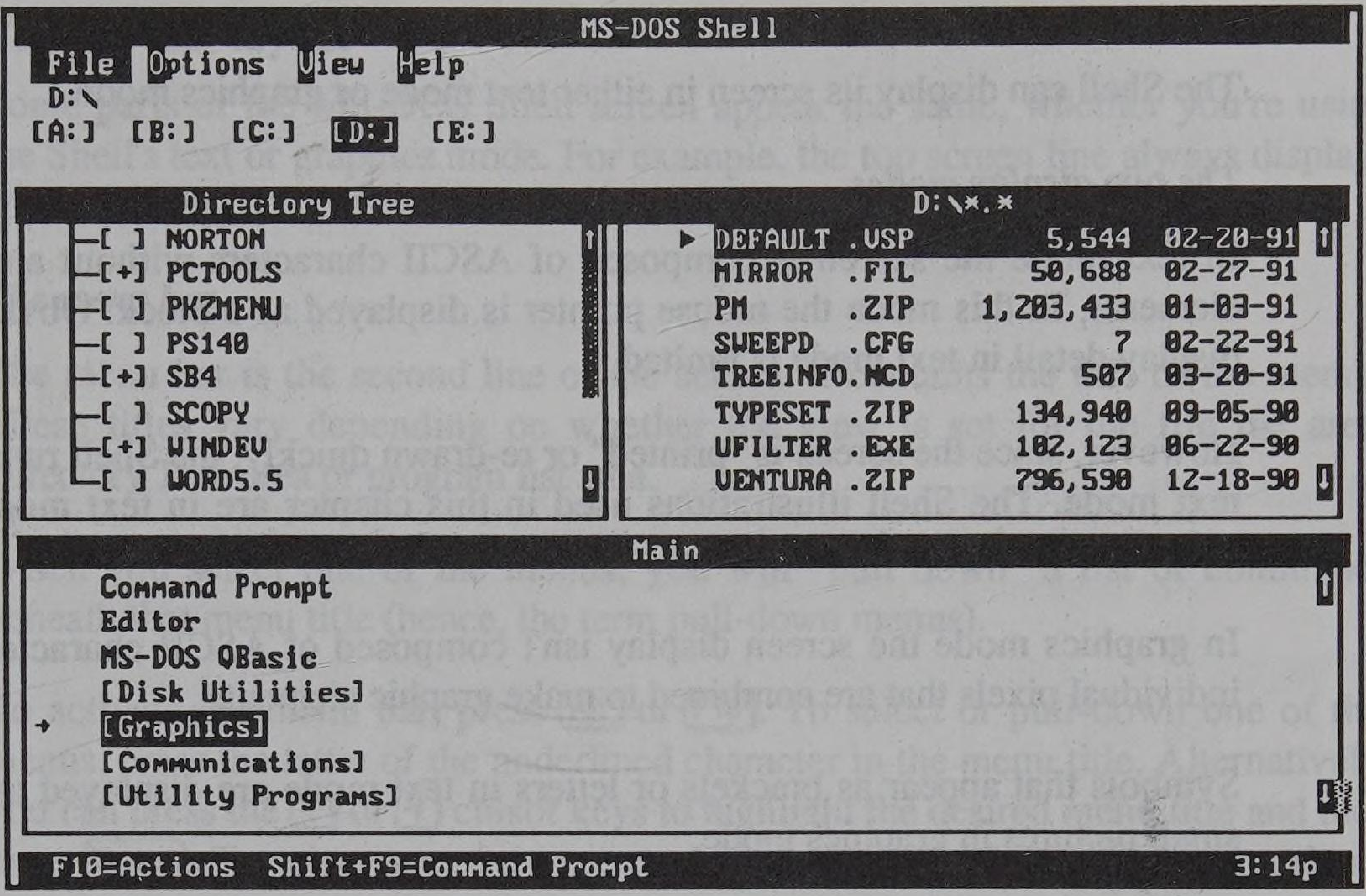
The Shell simplifies your work with files and directories. The Shell file list and Directory Tree areas present the directory and filenames of a floppy diskette or hard drive in one of several *views*.

By selecting one of the commands in the **View** menu, you can increase or decrease the amount of information on the screen. This decreases the amount of "clutter" on the screen. You can easily create and change directories or copy, compare, delete and rename files, for example.

Running programs from the MS-DOS Shell

The Shell also lets you start programs. The Shell's program list area displays all of the programs you can select and start. To start a program, select it using the cursor keys and press Enter or double-click the mouse button.

You can also group programs. When programs are grouped, the group name appears in the list rather than the names of the individual programs. Selecting the group displays the names of all programs belonging to this group. By using this method, you can organize the programs on your disk more effectively.



```
D:\
[A:]  [B:]  [C:]  [D:]  [E:]
        Directory Tree                          D:\*.*
   ├─[ ] NORTON                ▶ DEFAULT .VSP        5,544  02-20-91
   ├─[+] PCTOOLS                 MIRROR  .FIL       50,688  02-27-91
   ├─[ ] PKZMENU                 PM      .ZIP    1,203,433  01-03-91
   ├─[ ] PS140                   SWEEPD  .CFG            7  02-22-91
   ├─[+] SB4                     TREEINFO.NCD          507  03-20-91
   ├─[ ] SCOPY                   TYPESET .ZIP      134,940  09-05-90
   ├─[+] WINDEV                  VFILTER .EXE      102,123  06-22-90
   └─[ ] WORDS5.5                VENTURA .ZIP      796,590  12-18-90
                                 Main
      Command Prompt
      Editor
      MS-DOS QBasic
      [Disk Utilities]
  →   [Graphics]
      [Communications]
      [Utility Programs]
 F10=Actions  Shift+F9=Command Prompt                             3:14p
```

Program groups have been added to the program list area for user convenience

15.1.2 Opening the MS-DOS Shell

When you start the Shell with the DOSSHELL command, the screen that's displayed is quite different from the usual command line and system prompt. By pressing a few keys or by moving the mouse around and clicking, you can perform many operations without typing the DOS commands.

Using the mouse

It's very easy to operate the MS-DOS Shell with the mouse. Moving the mouse on your desk causes the mouse pointer to move on the screen. This lets you position the mouse pointer (a small rectangular block or arrow) over different commands, groups or items.

Pressing down the left mouse button selects a command. This is called "clicking". Pressing the mouse button twice is called "double-clicking".

Unlike selecting a command from the keyboard, double-clicking activates a command. Depending on the command, this action can have several effects.

The MS-DOS Shell Screen

The Shell can display its screen in either text mode or graphics mode.

The two display modes

In text mode the screen is composed of ASCII characters without any graphic elements. In this mode the mouse pointer is displayed as a block. Obviously, the display detail in text mode is limited.

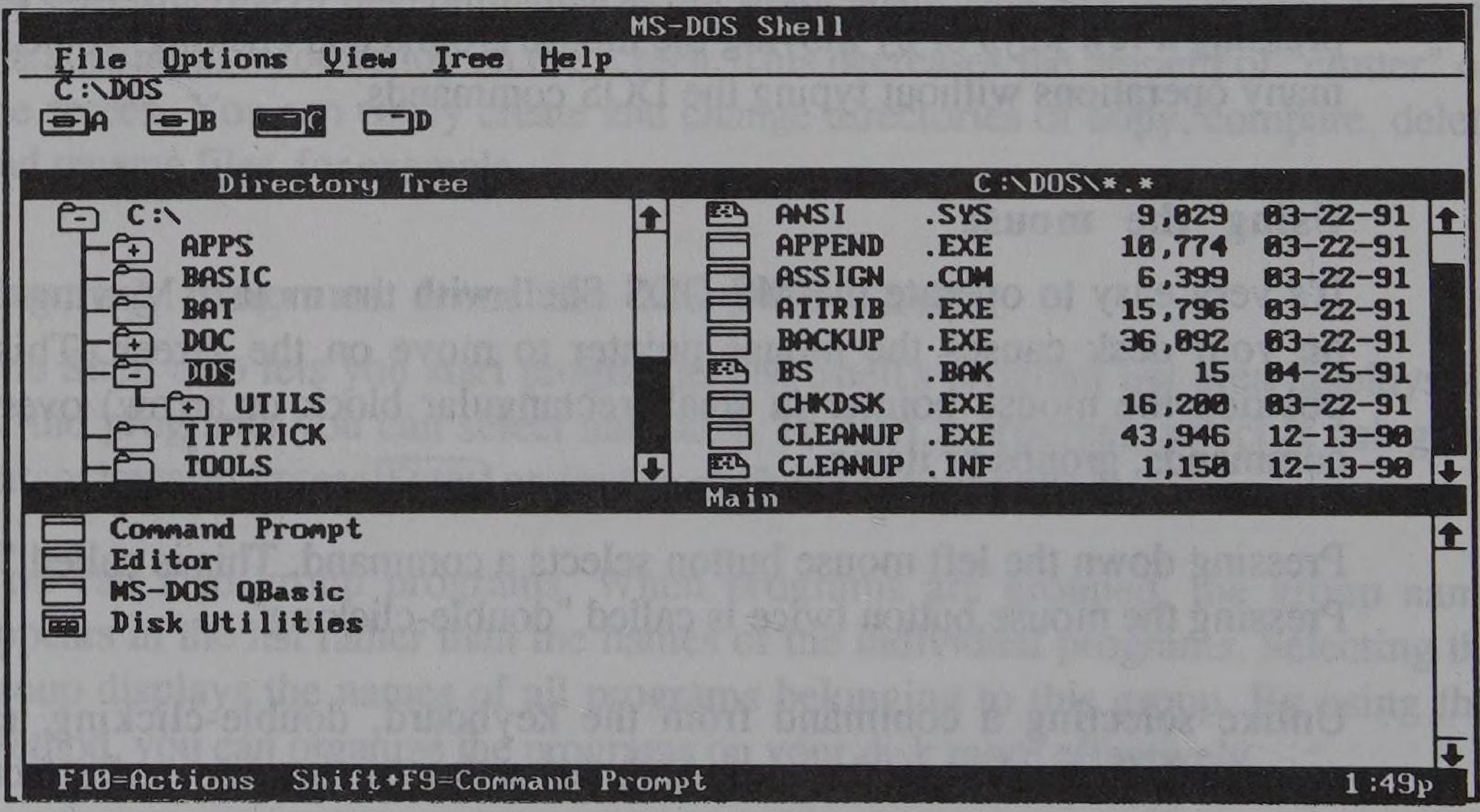
However, since the screen is "printed" or re-drawn quickly, the Shell runs faster in text mode. The Shell illustrations used in this chapter are in text mode (unless otherwise noted).

In graphics mode the screen display isn't composed of ASCII characters, but of individual pixels that are combined to make graphic elements.

Symbols that appear as brackets or letters in text mode are displayed as icons or small pictures in graphics mode.

In graphics mode the mouse pointer is displayed as a small diagonal arrow that changes in appearance under certain conditions. Although graphics mode gives you greater detail, the speed of the display is slower than in text mode.

The following illustration shows the MS-DOS Shell in graphics mode. Although this display differs slightly from the display in text mode, closer examination reveals that it contains the same basic information.


MS-DOS Shell
File Options View Tree Help
C:\DOS
A B C D
Directory Tree
C:\DOS\*.*
Main
Command Prompt
Editor
MS-DOS QBasic
Disk Utilities
F10=Actions Shift+F9=Command Prompt
1:49p

```
C:\
  APPS
  BASIC
  BAT
  DOC
  DOS
    UTILS
  TIPTRICK
  TOOLS
```

```
ANSI     .SYS     9,029  03-22-91
APPEND   .EXE    10,774  03-22-91
ASSIGN   .COM     6,399  03-22-91
ATTRIB   .EXE    15,796  03-22-91
BACKUP   .EXE    36,092  03-22-91
BS       .BAK        15  04-25-91
CHKDSK   .EXE    16,200  03-22-91
CLEANUP  .EXE    43,946  12-13-90
CLEANUP  .INF     1,150  12-13-90
```

The MS-DOS Shell in graphics mode

The screen layout

Some parts of the MS-DOS Shell screen appear the same, whether you're using the Shell's text or graphics mode. For example, the top screen line always displays "MS-DOS Shell".

The menu bar

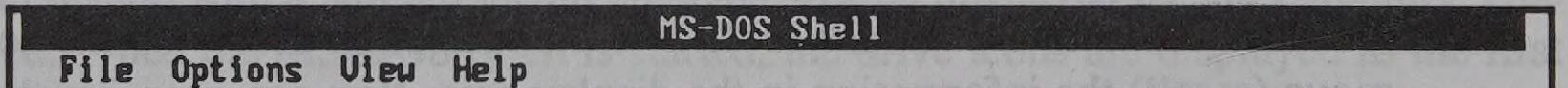
The menu bar is the second line of the screen. It contains the title of the menus. These titles vary depending on whether the view is set for the file list area, Directory Tree area or program list area.

When you select one of the menus, you will "pull down" a list of commands beneath that menu title (hence, the term pull-down menus).

To activate the menu bar, press Alt or F10. To select or pull-down one of the menus, press the letter of the underlined character in the menu title. Alternatively, you can press the ← or → cursor keys to highlight the desired menu title and then press Enter. To select one of the menus with the mouse, click on the menu title.

The title bar

Each area has its own title bar, which contains the title (the subject) of the area. For example, if the view is set to program list, the title bar contains the name of the active group. In the following example, the title bar contains "Main". This is the default group.


MS-DOS Shell
File Options View Help
Main
Command Prompt
Editor
MS-DOS QBasic
[Disk Utilities]
F10=Actions
Shift+F9=Command Prompt
4:06p

The MS-DOS Shell displaying only the program list area

The title bar displays the words "Directory Tree" above the area of the screen that displays the names of the subdirectories. Above the area of the screen that displays the file list, the title bar contains the full path of the current directory.

If more than one area of the screen is displayed, you can switch between the two areas by pressing the Tab key. To select one of the areas with the mouse, move the mouse pointer to the title bar of the desired area and click.

The scroll bars

Each area of the Shell has a vertical grey bar along its right border, which contains a direction arrow pointing up and down. This is the scroll bar. If an area contains more information than will fit in the area, a grey slider bar appears in the scroll bar.

The slider bar indicates the relative position of the information that is displayed. If the slider bar is positioned at the top of the scroll bar, then the top portion of the information is being displayed. If the bar is in the middle of the scroll bar, you are seeing the middle portion of the information and if it appears at the bottom of the scroll bar, the end of the information is displayed.

To move the contents of the area, press the ↑ and ↓ cursor keys or the PgUp and PgDn keys. Notice that the scroll bar moves as you press the keys. Scroll bars not only help you move within an area, but also help you get your bearings within the area. They are an important component of the SAA concept.

You can also use the mouse to click either direction arrow of the scroll bar to move (scroll) the information in the display area. You can even use the mouse to move the bar directly. Place the mouse pointer on the bar, press the left mouse button and, while holding it down, move the mouse pointer in the desired direction. This process is referred to as "dragging". The scroll bar will move along with the mouse arrow to display the corresponding information in the display area.

The status bar

The last line of the screen is called the status bar. Here the Shell displays the purpose of the most important function keys and the time. F10 activates the menu bar and Shift + F9 can be used to exit the MS-DOS Shell to enter commands on the MS-DOS command line.

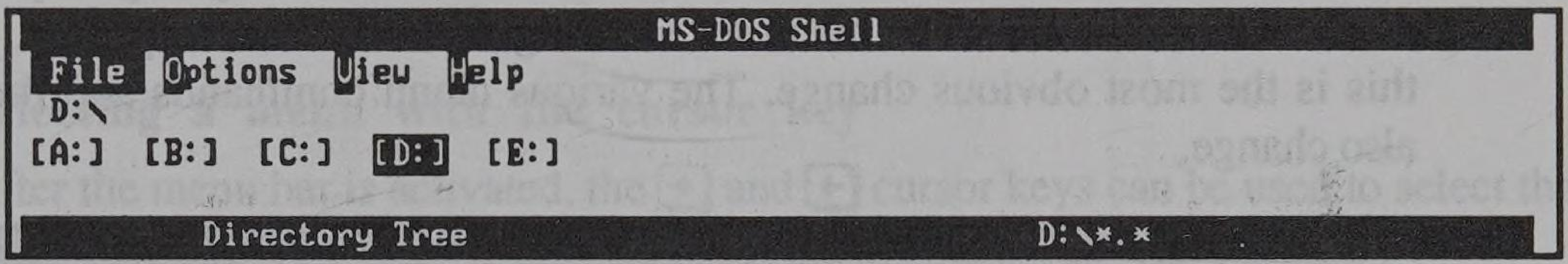
You can later return to the MS-DOS Shell by typing "EXIT". The right corner of the status bar displays the current time or system time. Occasionally messages and prompts also appear on this line (e.g., "Insert diskette in drive B: and press any key").

15.2 MS-DOS Shell Fundamentals

Since the MS-DOS Shell is based on the SAA standard, the individual work areas operate in similar ways. So we'll only summarize the basic processes.

15.2.1 Working in an area

As we mentioned in the introduction, after starting the MS-DOS Shell, the file list area, the Directory Tree area and the program list area are displayed. The logical drives of your PC are displayed between the menu bar and the title bars of the file list area and Directory Tree area.



List of logical drives in the MS-DOS Shell

The logical drives

This line is referred to as the "drive icons" and is always on the screen when **Program/File List** from the **View** menu is selected. By selecting the drive you can decide which storage media contents will be displayed by the Directory Tree area. When the MS-DOS Shell is started, the drive icons are displayed as the first area.

To select a new drive, use the → and ← keys to select the appropriate icon and activate it with the Spacebar or by pressing Enter. You can also change the active drive by pressing Ctrl + drive letter. For example, press Ctrl + A to change the current drive to drive A:. Pressing Ctrl + C changes the current drive to drive C:.

The mouse can be used to click directly on the desired drive icon to activate it and to display its Directory Tree and file list areas.

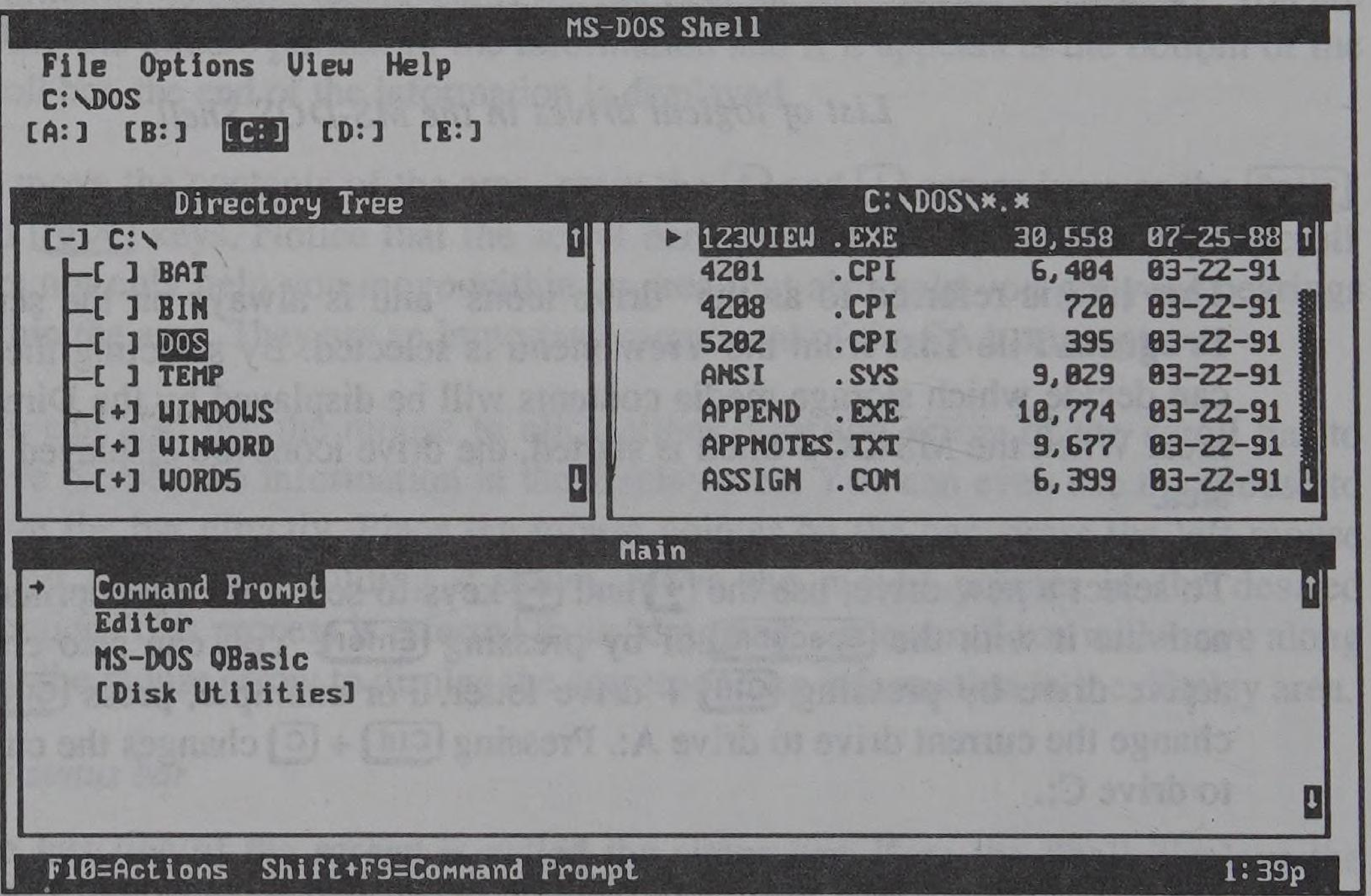
Double-clicking the left mouse button while pointing to a drive icon causes the file information to be read again. This is useful after you've changed a diskette in the selected drive. You can also force the selected drive to be re-read from the keyboard by selecting the drive icon and pressing Enter.

Changing areas with Tab

After selecting the desired drive, use Tab to change to the next area. Each time Tab is pressed the active area changes from right to left and from top to bottom. Pressing Shift + Tab changes the areas in reverse order. This enables you to move the selection cursor to the desired area very quickly.

Using the mouse is an even faster way to activate an area. Point to the title bar of an area and click the left mouse button. You can also click any item within the area to activate that area. This process also selects the item within the area. However, you may not want to select this item.

When changing from either the Directory Tree area or file list area to the program list area (or vice versa), the entries in the menu bar are changed. The program list area doesn't have the **Tree** menu for working with the Directory Tree. However, this is the most obvious change. The various menu commands and their effects also change.



```
                              MS-DOS Shell
 File  Options  View  Help
 C:\DOS
[A:]  [B:]  [C:]  [D:]  [E:]

          Directory Tree                          C:\DOS\*.*
[-] C:\                              ► 123VIEW .EXE      30,558  07-25-88
  ├─[ ] BAT                             4201    .CPI       6,404  03-22-91
  ├─[ ] BIN                             4208    .CPI         720  03-22-91
  ├─[ ] DOS                             5202    .CPI         395  03-22-91
  ├─[ ] TEMP                            ANSI    .SYS       9,029  03-22-91
  ├─[+] WINDOWS                         APPEND  .EXE      10,774  03-22-91
  ├─[+] WINWORD                         APPNOTES.TXT       9,677  03-22-91
  └─[+] WORD5                           ASSIGN  .COM       6,399  03-22-91

                                   Main
 → Command Prompt
   Editor
   MS-DOS QBasic
   [Disk Utilities]

 F10=Actions  Shift+F9=Command Prompt                                1:39p
```

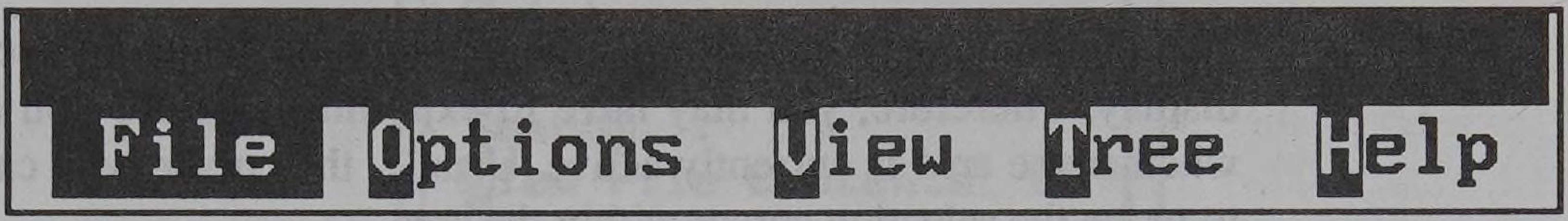
MS-DOS Shell with the program list area activated

15.2.2 Using the pull-down menus

The menu bar contains the names of the pull-down menus. A menu contains a list of commands that can be selected and activated.

Pressing Alt or F10 activates the menu bar from the keyboard.

You can tell that the menu bar is activated because "File" appears in inverse video (the background changes color). Also, each title in the menu bar either has its first letter underlined (graphics mode), has a colored mark (text mode in color) or has a selection cursor (monochrome text mode).



Menu bar activated in low resolution text mode

There are different methods for opening one of these pull-down menus and accessing the desired command.

Selecting a menu with the cursor key

After the menu bar is activated, the → and ← cursor keys can be used to select the desired menu. Then use the ↑ or ↓ cursor keys to open the menu.

When the menu opens, use the ↑ or ↓ cursor keys to select one of the commands. Press Enter to activate a command.

Selecting a menu with the letter keys

Another, faster option for selecting a menu consists of pressing the letter keys. Just press the first letter in the name and the pull-down menu will open. You can also use the letter keys to select commands.

After opening the menu, press the highlighted letter of the activated command to perform the function.

Selecting a menu with the mouse

Another way to select a menu is by using the mouse. Click on the desired menu title to open the pull-down menu. You can also click on each command in the open menu.

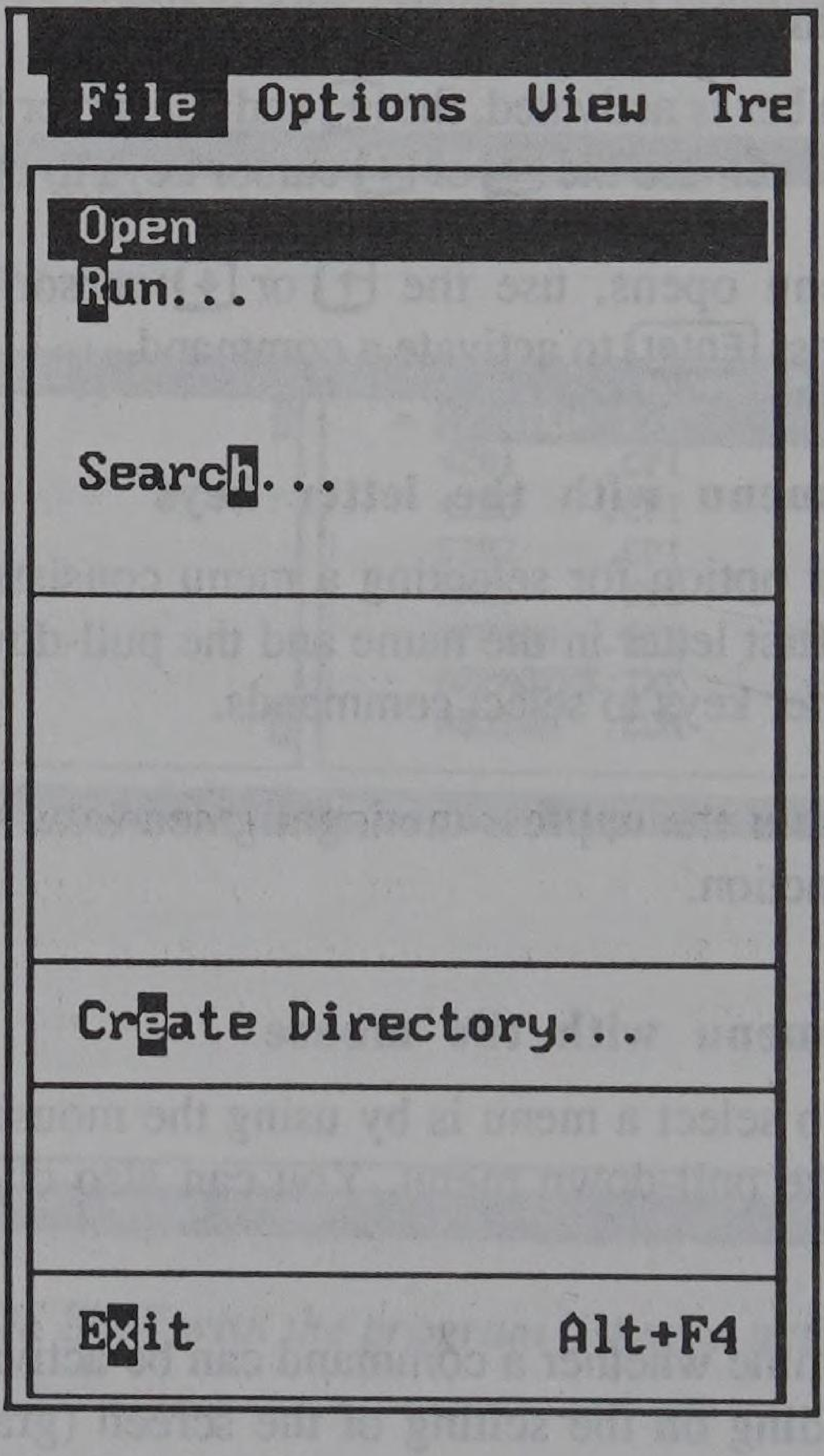
You can determine whether a command can be activated by its appearance on the screen. Depending on the setting of the screen (graphics or text) and the color (color or monochrome), the commands that cannot be activated are either invisible or dimmed (have a colored or grey background).

We'll discuss how to use the **Colors...** and **Display...** commands of the **Options** menu later in this chapter.

The menu titles and commands change when you switch among the file list area, Directory Tree area and the program list area. Usually, the area that is currently active is highlighted in the title bar.

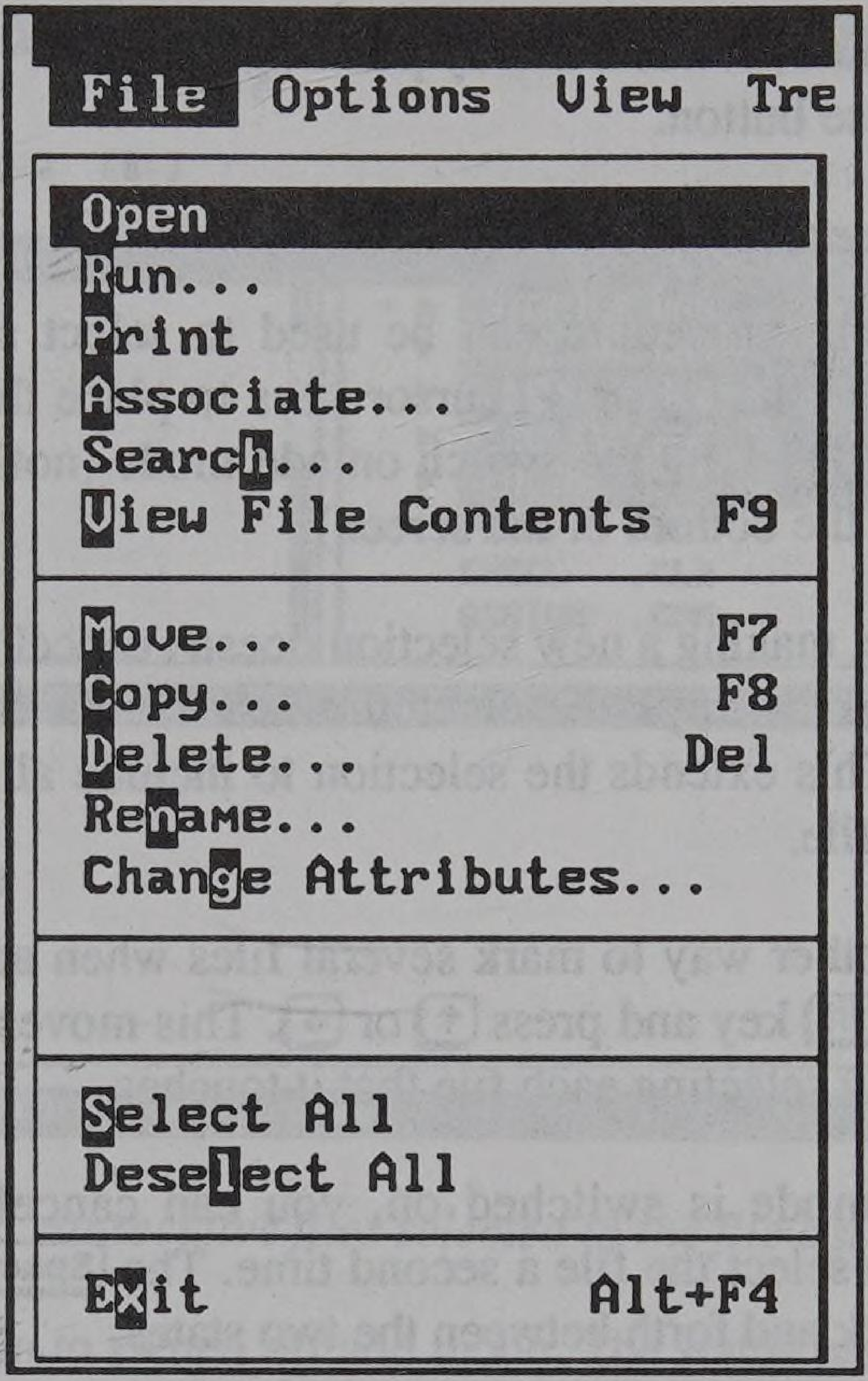
The highlight may occasionally be difficult to detect in every screen display. Therefore, you may have to experiment before you determine whether the area is currently active. Usually the cursor keys can be used to move the selection cursor within the area.

For example, as you can see in the following picture, the **File** menu has five activated commands when the Directory Tree area is active. You cannot activate a command such as **Print** or **View File Contents** because these commands are dimmed.



File menu with Directory Tree area active

However, when you activate the file list area (by pressing Tab until the area is highlighted), many more commands are active in the **File** menu:


File Options View Tre
Open
Run...
Print
Associate...
Search...
View File Contents F9
Move... F7
Copy... F8
Delete... Del
Rename...
Change Attributes...
Select All
Deselect All
Exit Alt+F4

File menu with file list area active

15.2.3 Selecting files and other elements

The commands of the pull-down menus (especially those of the **File** menu) only work when you use them on an element such as a group, a program, a directory or a file. So you must select one of these elements before the command is activated. For example, to copy a file, you must first select the file.

Selecting files

Use either the cursor keys or the mouse to select the directory in the left area of the screen (the Directory Tree area). Notice that as you select different directories, files in that directory appear in the file list area (right area of the screen).

Use Tab to switch to the file list area, where you can select files to be edited. Use either the ↑ or ↓ cursor keys or the mouse and the scroll bar to select the file.

Selecting one file

You can select a file from the keyboard by using the ↑ and ↓ cursor keys to move the selection cursor to the desired file. A filename is selected when it appears in inverse video.

With the mouse, select a file by pointing to it with the mouse pointer and clicking the left mouse button.

Selecting several consecutive files

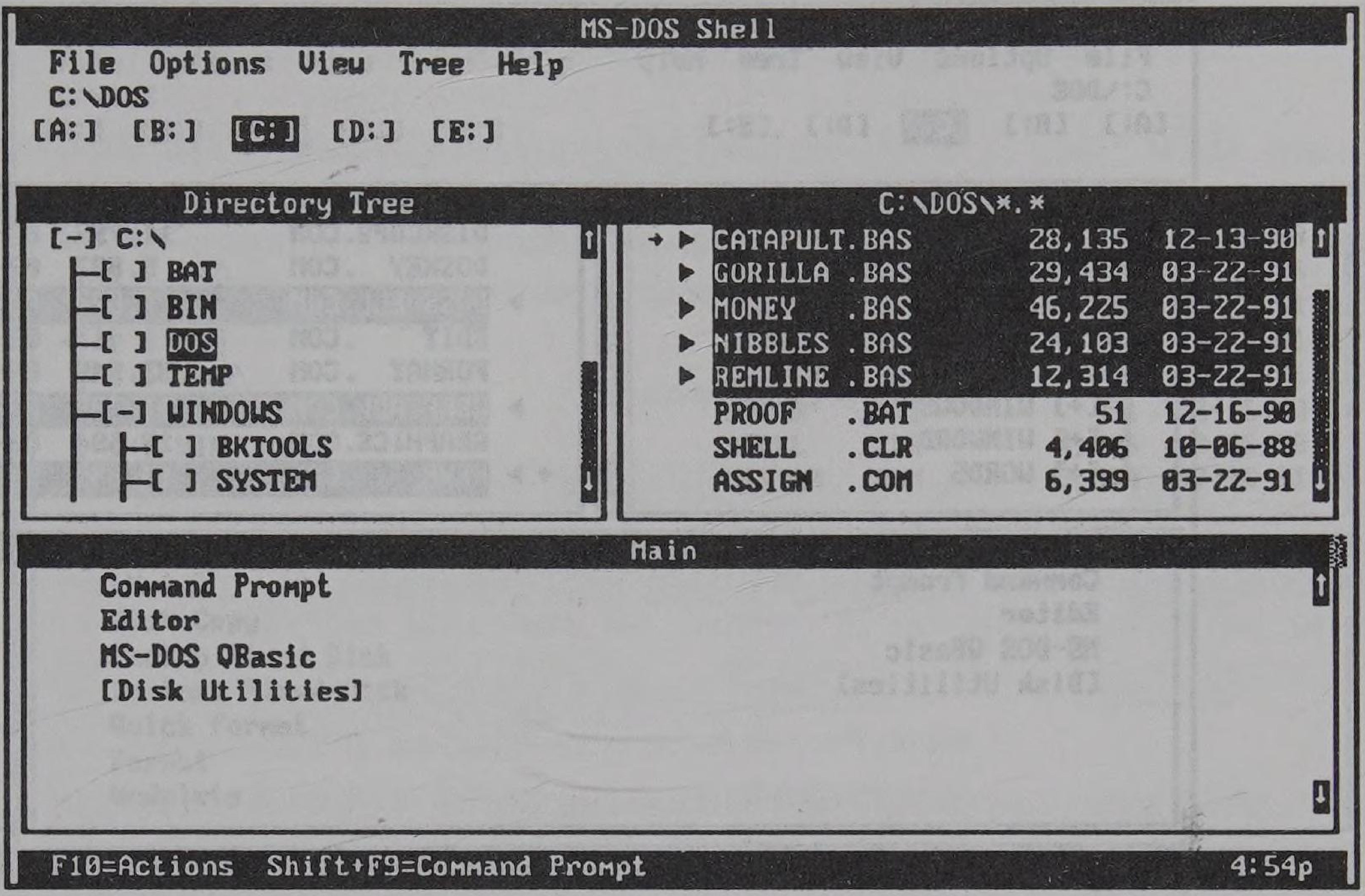
The following procedure can be used to select several consecutive files in a directory. Use the ↑ or ↓ cursor keys to place the selection cursor on the first file. Press Shift + F8 to switch on add mode (notice that "ADD" appears in the status bar at the bottom of the screen).

In add mode, making a new selection doesn't cancel the previous selection. Use the ↑ or ↓ cursor keys to select the last file in the series and press Shift + Spacebar. This extends the selection to include all of the files between the first and the last file.

There is another way to mark several files when add mode is switched on. Hold down the Shift key and press ↑ or ↓. This moves the selection through the list, automatically selecting each file that it touches.

When add mode is switched on, you can cancel a selection by pressing the Spacebar to select the file a second time. The Spacebar works like a switch that switches back and forth between the two states.

Selecting several files in the MS-DOS Shell is similar to using wildcards in the command line. Although the only way to handle several files from the command line is if they have the same name (BOOK.* or *.EXE), the MS-DOS Shell is much more flexible. The files can be displayed in different ways. However, the displayed sequence of files in the MS-DOS Shell can be specified alphabetically, according to file extension, etc. This assures that the desired files appear together.



```
                               MS-DOS Shell
 File  Options  View  Tree  Help
 C:\DOS
[A:]  [B:]  [C:]  [D:]  [E:]

       Directory Tree                          C:\DOS\*.*
 [-] C:\                        → ▶ CATAPULT.BAS     28,135  12-13-90
  ├─[ ] BAT                       ▶ GORILLA .BAS     29,434  03-22-91
  ├─[ ] BIN                       ▶ MONEY   .BAS     46,225  03-22-91
  ├─[ ] DOS                       ▶ NIBBLES .BAS     24,103  03-22-91
  ├─[ ] TEMP                      ▶ REMLINE .BAS     12,314  03-22-91
  └─[-] WINDOWS                     PROOF   .BAT         51  12-16-90
      ├─[ ] BKTOOLS                 SHELL   .CLR      4,406  10-06-88
      └─[ ] SYSTEM                  ASSIGN  .COM      6,399  03-22-91

                                   Main
     Command Prompt
     Editor
     MS-DOS QBasic
     [Disk Utilities]

 F10=Actions  Shift+F9=Command Prompt                           4:54p
```

Several consecutive files selected in the file list area

Press Shift + F8 again to switch off add mode. The status bar changes back to its original configuration.

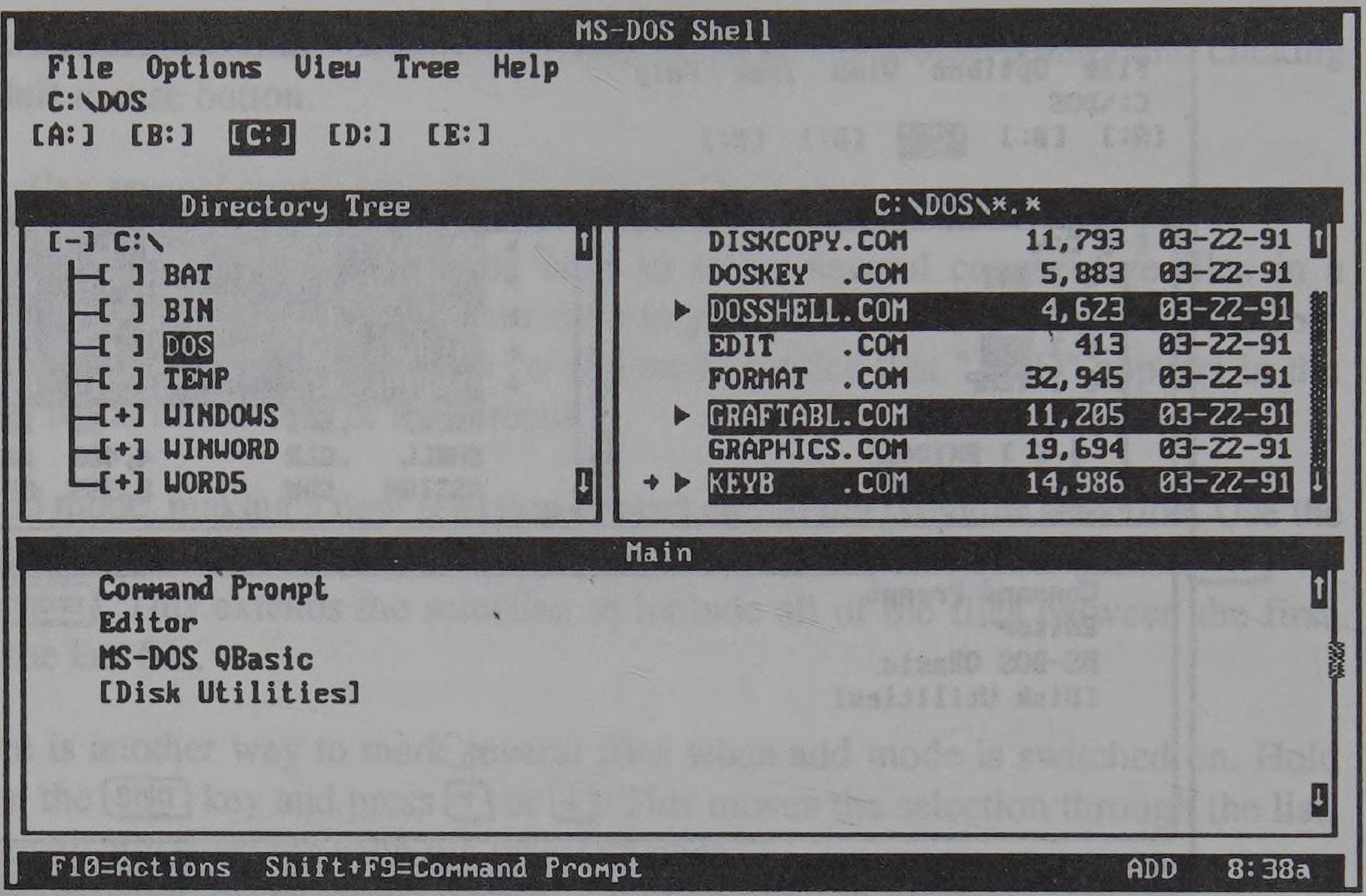
To select multiple files with the mouse, click the first file, hold down the Shift key, click the last file and release Shift. This selects the first and last file, in addition to all files in between. To cancel the selection of the last file, hold down the Shift key and click the scroll bar.

Selecting multiple non-consecutive files

Use the ↑ or ↓ cursor keys to select the first file. Press Shift + F8 to activate add mode. Use the cursor keys again to choose the next file and press the Spacebar to select it. Continue this process until all desired files are selected. To cancel the selection of a file, use the cursor keys to select it and press the Spacebar again.

Press Shift + F8 to switch add mode off again.

If you are using a mouse, click on the first file to select it. Then hold down the Ctrl key and point and click on the next file to select it. Continue until all of the necessary files have been selected. Release Ctrl. To move within the area, you can also hold down Ctrl and click the scroll bar.



```
                              MS-DOS Shell
 File  Options  View  Tree  Help
 C:\DOS
 [A:]  [B:]  [C:]  [D:]  [E:]
          Directory Tree                         C:\DOS\*.*
 [-] C:\                         ↑       DISKCOPY.COM        11,793  03-22-91 ↑
   ├[ ] BAT                              DOSKEY  .COM         5,883  03-22-91
   ├[ ] BIN                            ▶ DOSSHELL.COM         4,623  03-22-91
   ├[ ] DOS                              EDIT    .COM           413  03-22-91
   ├[ ] TEMP                             FORMAT  .COM        32,945  03-22-91
   ├[+] WINDOWS                        ▶ GRAFTABL.COM        11,205  03-22-91
   ├[+] WINWORD                          GRAPHICS.COM        19,694  03-22-91
   └[+] WORDS                    ↓   → ▶ KEYB    .COM        14,986  03-22-91 ↓
                                    Main
       Command Prompt                                                         ↑
       Editor
       MS-DOS QBasic
       [Disk Utilities]
                                                                              ↓
 F10=Actions  Shift+F9=Command Prompt                            ADD   8:38a
```

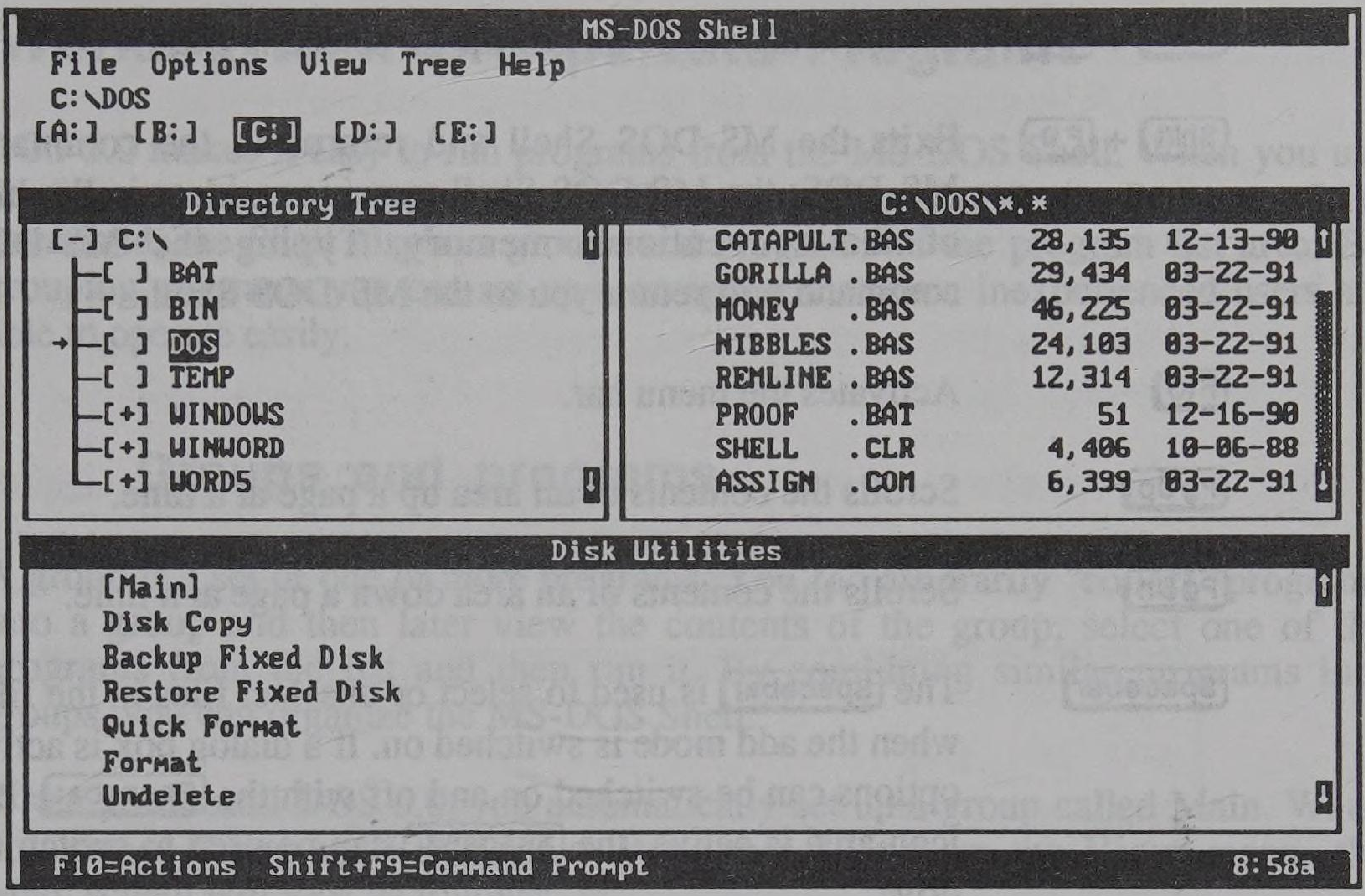
Several non-consecutive files selected in the file list area

Selecting groups and programs

There's an even easier way to select files. You can select groups of files or programs defined in the program list area. Use either the ↑ or ↓ cursor keys or click the mouse once to select the group or program. You can also use the commands from the pull-down menu.

To start a program, either press Enter or double-click on the program name in the file list or program list areas. Double-clicking on a group in the program list area causes the contents of the group to be displayed.

Each group has a menu command called Main, which can be used to return to the main group. Pressing Esc moves one level higher within nested groups.



```
                              MS-DOS Shell
 File  Options  View  Tree  Help
 C:\DOS
[A:]  [B:]  [C:]  [D:]  [E:]
        Directory Tree                          C:\DOS\*.*
 [-] C:\                          CATAPULT.BAS        28,135  12-13-90
   ├[ ] BAT                       GORILLA .BAS        29,434  03-22-91
   ├[ ] BIN                       MONEY   .BAS        46,225  03-22-91
  →├[ ] DOS                       NIBBLES .BAS        24,103  03-22-91
   ├[ ] TEMP                      REMLINE .BAS        12,314  03-22-91
   ├[+] WINDOWS                   PROOF   .BAT            51  12-16-90
   ├[+] WINWORD                   SHELL   .CLR         4,406  10-06-88
   └[+] WORDS                     ASSIGN  .COM         6,399  03-22-91
                             Disk Utilities
     [Main]
     Disk Copy
     Backup Fixed Disk
     Restore Fixed Disk
     Quick Format
     Format
     Undelete
 F10=Actions  Shift+F9=Command Prompt                              8:58a
```

The Disk Utilities group open in the program list area; note the command [Main] for returning to the top level

How the keyboard operates in the MS-DOS Shell

The following is an overview of the functions provided by the keyboard in the MS-DOS Shell:

Enter	Activates a selected command, starts a selected program or confirms an entry. When a drive icon is selected, pressing Enter reads the directory structure of the drive again.
Esc	Use this key to terminate a selected menu or a dialog box. If you are in a group, pressing Esc takes you to the next highest group.
Alt	Activates the menu bar.
F1	Use this key to call for help in the MS-DOS Shell. The MS-DOS Shell displays the help screen of the currently selected command.
F2	Pressing F2 copies programs to another group.
F3	Pressing F3 exits the MS-DOS Shell and returns to the MS-DOS command line at the system prompt.
Alt + F4	Leaves the MS-DOS Shell and returns to the command line of MS-DOS.

Shift + F8	Switches add mode on and off for selecting files.
Shift + F9	Exits the MS-DOS Shell and returns to the command line of MS-DOS; the MS-DOS Shell remains resident in the background of the conventional memory. Typing the MS-DOS EXIT command will return you to the MS-DOS Shell.
F10	Activates the menu bar.
PgUp	Scrolls the contents of an area up a page at a time.
PgDn	Scrolls the contents of an area down a page at a time.
Spacebar	The Spacebar is used to select or unselect files in the file list area when the add mode is switched on. If a dialog box is active, certain options can be switched on and off with the Spacebar. If the drive icon area is active, the Spacebar can be used to switch to another drive.
Tab	Use Tab to move between areas of the MS-DOS Shell. If a dialog box is active, press Tab to move to the next dialog item. Shift + Tab moves the cursor to the previous item.

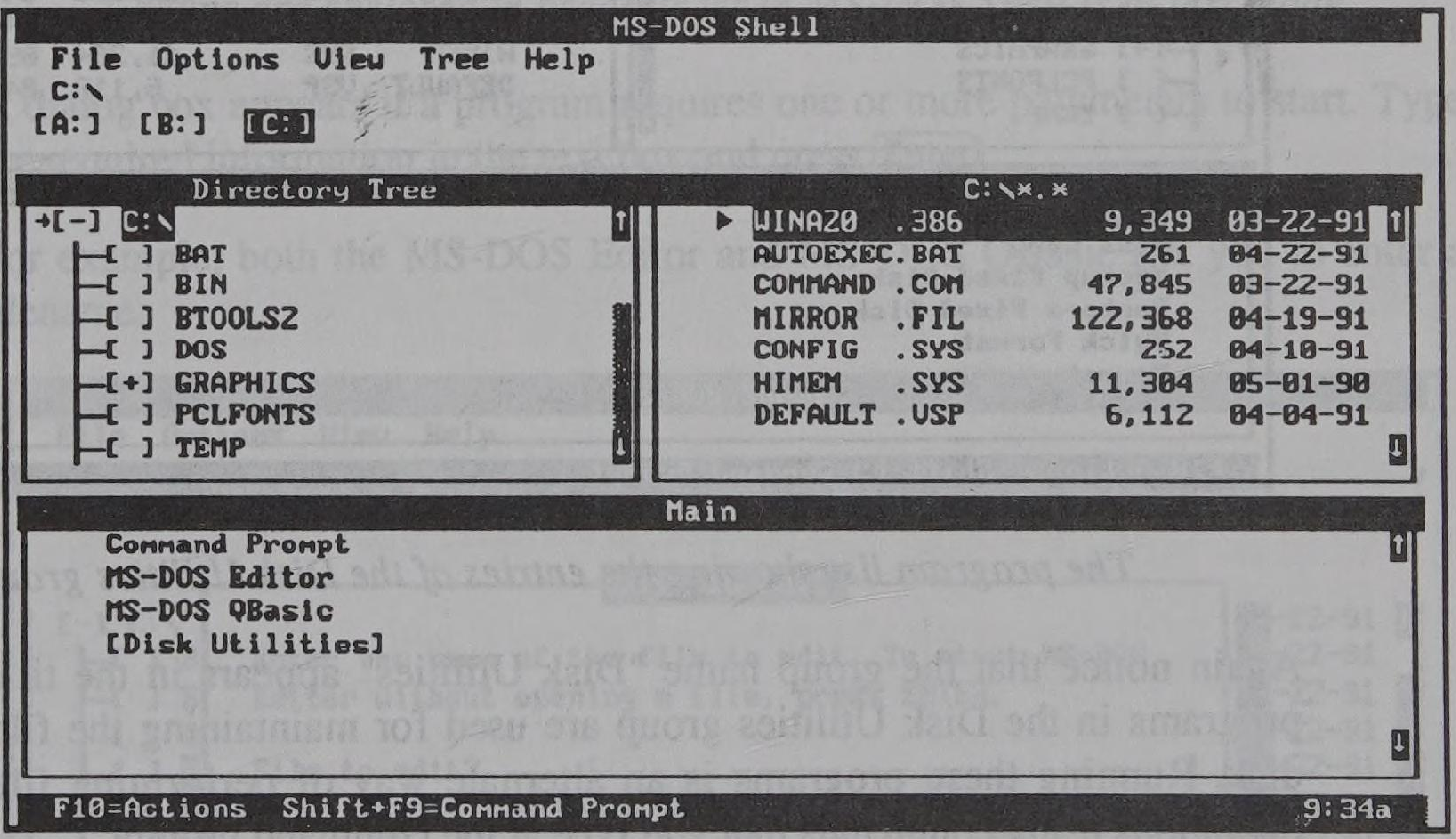
15.3 Working With Groups And Programs

DOS 5.0 makes it easy to run programs from the MS-DOS Shell. When you use the **View** menu to select either the **Program/File List** or the **Program List** command, the Shell displays groups and programs in the program list area. By grouping programs you can set up a computer that even inexperienced users are able to operate easily.

15.3.1 Groups and programs

A group is a set of one or more programs. You can arbitrarily "collect" programs into a group and then later view the contents of the group, select one of the programs from the list and then run it. By combining similar programs into groups, you can organize the MS-DOS Shell.

When you install DOS 5.0, you automatically set up a group called Main. When you select **Program/File List** or **Program List** from the **View** menu, the Main screen will look as follows:



```
                                MS-DOS Shell
 File  Options  View  Tree  Help
 C:\
[A:]  [B:]  [C:]

          Directory Tree                               C:\*.*
→[-] C:\                                 ▶ WINA20   .386      9,349  03-22-91
  ├[ ] BAT                                 AUTOEXEC.BAT        261  04-22-91
  ├[ ] BIN                                 COMMAND .COM     47,845  03-22-91
  ├[ ] BTOOLSZ                             MIRROR  .FIL    122,368  04-19-91
  ├[ ] DOS                                 CONFIG  .SYS        252  04-10-91
  ├[+] GRAPHICS                            HIMEM   .SYS     11,304  05-01-90
  ├[ ] PCLFONTS                            DEFAULT .USP      6,112  04-04-91
  └[ ] TEMP
                                     Main
     Command Prompt
     MS-DOS Editor
     MS-DOS QBasic
     [Disk Utilities]

 F10=Actions  Shift+F9=Command Prompt                                   9:34a
```

The program list showing the entries of the Main group

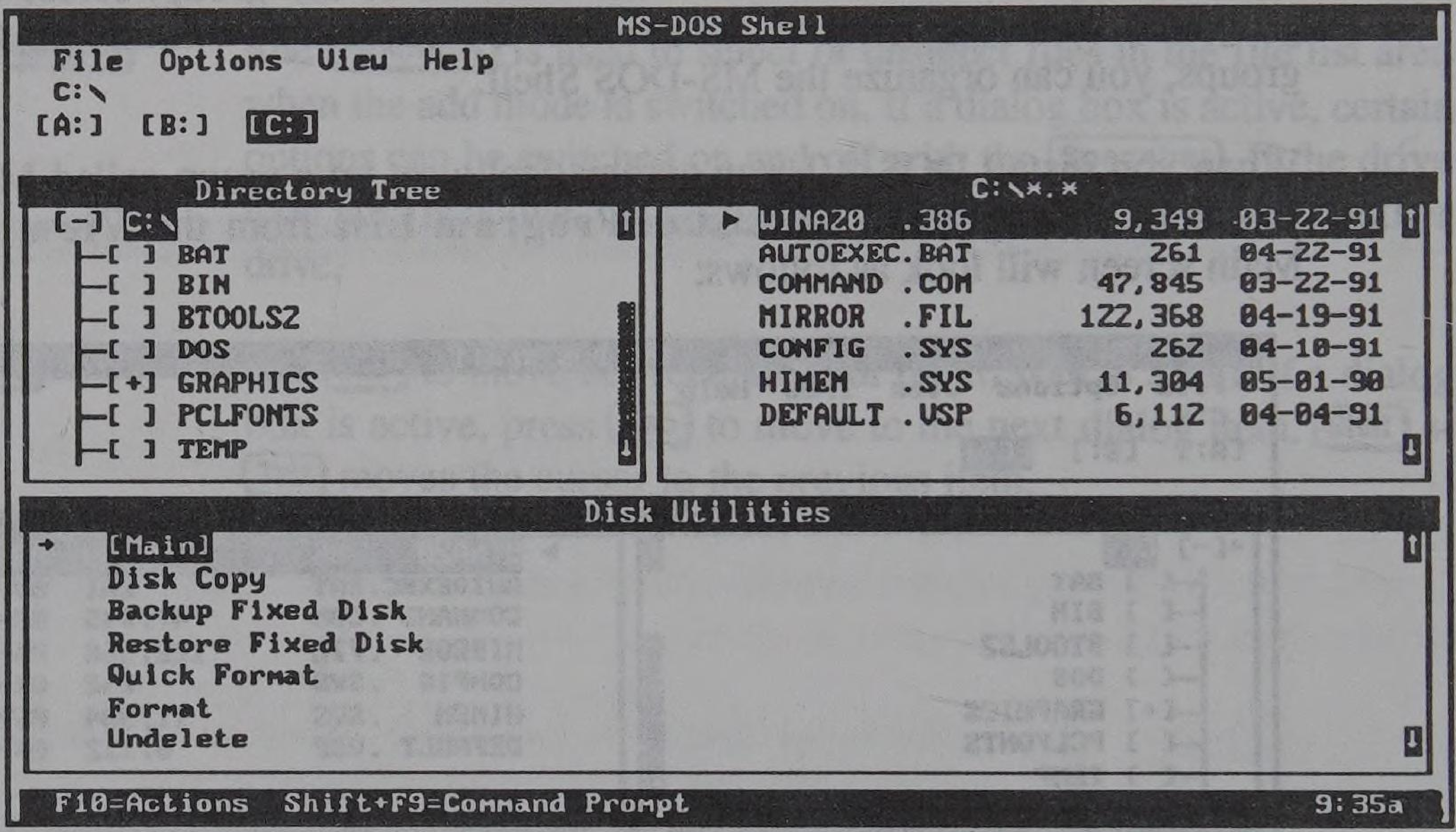
Notice that the group name is in the title bar of the program list area. You can see from the list that this group contains four entries: Command Prompt, Editor, MS-DOS QBasic and [Disk Utilities].

Since the first three entries are programs, they're listed without brackets. They are the command line system prompt (which isn't really a program but a way to enter a DOS command from the Shell), the MS-DOS Editor program EDIT and the QBasic programming language.

Since the last entry is enclosed within brackets, it must be the name of another group. This group is called the Disk Utilities group.

All you need to do to run a program is to select a program entry. When a program is run, the MS-DOS Shell "goes to sleep" and the selected program is started. You interact with the program just as if you had started it from the DOS command line. When you exit from that program, you return to the MS-DOS Shell which then "wakes up" again.

If you select a group entry, then the contents of that group are displayed in the program list area. So if you select the [Disk Utilities] group, you would see the following on the screen:



```
                              MS-DOS Shell
 File  Options  View  Help
 C:\
[A:]  [B:]  [C:]
            Directory Tree                               C:\*.*
 [-] C:\                                  ▶ WINA20   .386      9,349  03-22-91
   ├─[ ] BAT                                AUTOEXEC.BAT         261  04-22-91
   ├─[ ] BIN                                COMMAND .COM      47,845  03-22-91
   ├─[ ] BTOOLSZ                            MIRROR  .FIL     122,368  04-19-91
   ├─[ ] DOS                                CONFIG  .SYS         262  04-10-91
   ├─[+] GRAPHICS                           HIMEM   .SYS      11,304  05-01-90
   ├─[ ] PCLFONTS                           DEFAULT .VSP       6,112  04-04-91
   └─[ ] TEMP
                              Disk Utilities
 →   [Main]
     Disk Copy
     Backup Fixed Disk
     Restore Fixed Disk
     Quick Format
     Format
     Undelete
 F10=Actions  Shift+F9=Command Prompt                                   9:35a
```

The program list showing the entries of the Disk Utilities group

Again notice that the group name "Disk Utilities" appears in the title bar. The programs in the Disk Utilities group are used for maintaining the files on your disk. Running these programs is an alternate way of performing file and disk operations using commands that you type at the command prompt.

You'll see an entry for the Main group at the top of the list. By selecting Main, you can move back to the next highest group.

15.3.2 Starting programs

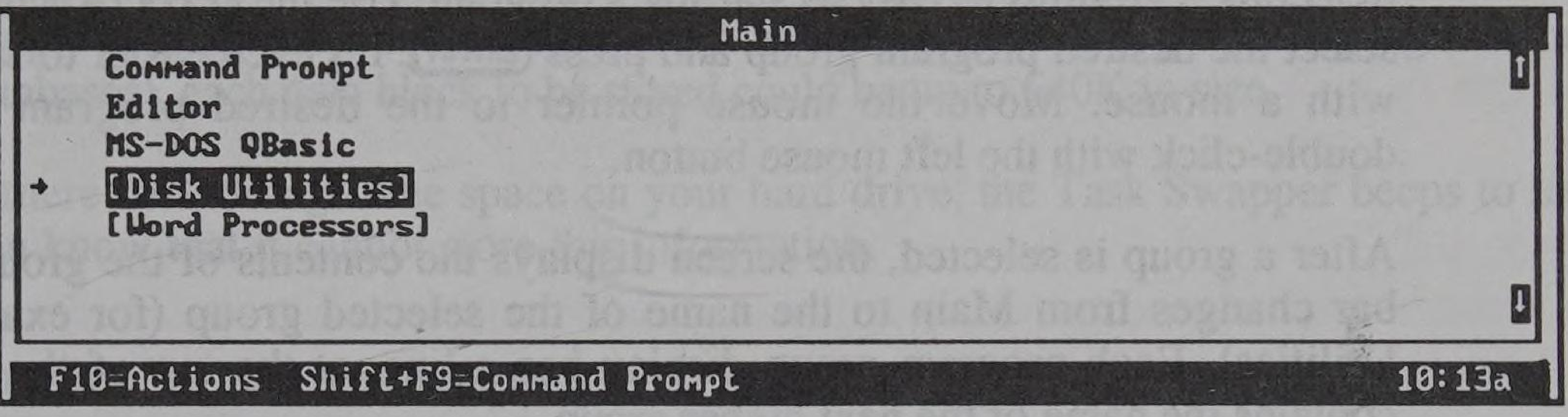
It's very easy to start programs from the program list area. Use the ↑ or ↓ cursor keys to select the desired program entry and press Enter.

Selecting and running a program is even easier with a mouse. Move the mouse pointer to the program entry and double-click the left mouse button.

The program icon

In the graphics mode of the MS-DOS Shell program, entries are displayed with an icon to the left of the entry. This icon resembles a box with a horizontal line near the top. A group entry will have an icon with six small rectangles within the icon.

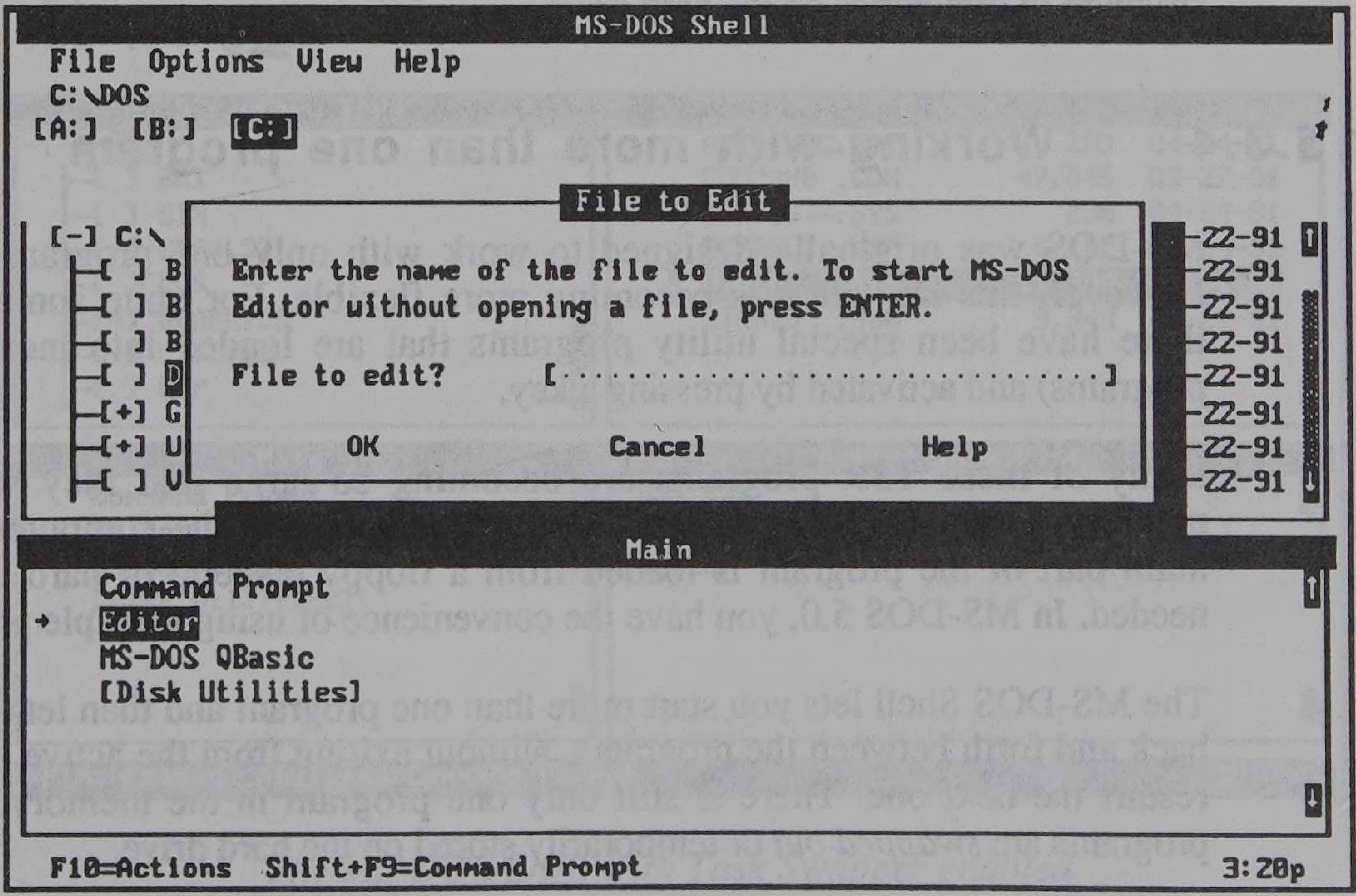
In text mode, groups in the program list are enclosed in brackets, as shown below:


Main
Command Prompt
Editor
MS-DOS QBasic
→ [Disk Utilities]
[Word Processors]
F10=Actions Shift+F9=Command Prompt
10:13a

Groups are enclosed in brackets when MS-DOS Shell is in text mode

A dialog box appears if a program requires one or more parameters to start. Type the required information in the text box and press (Enter).

For example, both the MS-DOS Editor and MS-DOS QBasic ask you to enter a filename.


MS-DOS Shell
File Options View Help
C:\DOS
[A:] [B:] [C:]
File to Edit
[-] C:\
[ ] B
[ ] B
[ ] B
[ ] D
[+] G
[+] U
[ ] V
-22-91
OK
Cancel
Help
Main
Command Prompt
→ Editor
MS-DOS QBasic
[Disk Utilities]
F10=Actions Shift+F9=Command Prompt
3:20p

Enter the name of the file to edit. To start MS-DOS Editor without opening a file, press ENTER.

File to edit? [.....................................]

The MS-DOS Editor dialog box

15.3.3 Selecting groups

A menu line containing a group of programs is called a program group.

The group icon

In graphics mode, a group appears as an icon to the left of the group name. The icon is a rectangle with six small rectangles inside. In text mode, a group appears with its name in brackets.

Selecting a group is as easy as starting a program. Use the ↑ or ↓ cursor keys to select the desired program group and press Enter. It's even easier to start a group with a mouse. Move the mouse pointer to the desired program group and double-click with the left mouse button.

After a group is selected, the screen displays the contents of the group. The title bar changes from Main to the name of the selected group (for example, Disk Utilities). Each program group display has a line, at the top of the menu, that contains the name of the next higher group.

If the next higher group is the Main group, then "Main" is displayed. Pressing Esc or double-clicking on the top line of each group takes you to the next higher group level. The ↑ or PgUp keys can also be used to select the entry. Press Enter to activate your selection.

A group can include an almost unlimited number of entries. These entries can be either programs or additional groups. The group structure slightly resembles the structure of directories on the hard drive.

15.3.4 Working with more than one program

MS-DOS was originally designed to work with only one program at a time. However, this limitation is becoming more flexible. For quite some time now there have been special utility programs that are loaded into memory (TSR programs) and activated by pressing a key.

Many of these TSR programs are becoming so large that they won't fit in memory. Often only a small portion remains active in the computer while the main part of the program is loaded from a floppy diskette or hard drive when needed. In MS-DOS 5.0, you have the convenience of using multiple programs.

The MS-DOS Shell lets you start more than one program and then lets you switch back and forth between the programs, without exiting from the active program, to restart the next one. There is still only one program in the memory. The other programs are *swapped out* or temporarily stored on the hard drive.

Although this method isn't as fast or convenient as a full multitasking operating system (e.g., Windows or DeskView), it does make some tasks easier.

The Task Swapper

The Task Swapper lets you switch between programs using the MS-DOS Shell. It keeps track of all of the current settings of the active program and temporarily stores the information on the hard drive when you switch to another program or back to the MS-DOS Shell.

This means everything the program has stored in the conventional memory area is temporarily saved on the hard drive as a hidden file. Your hard drive must have sufficient unused storage space available for the information.

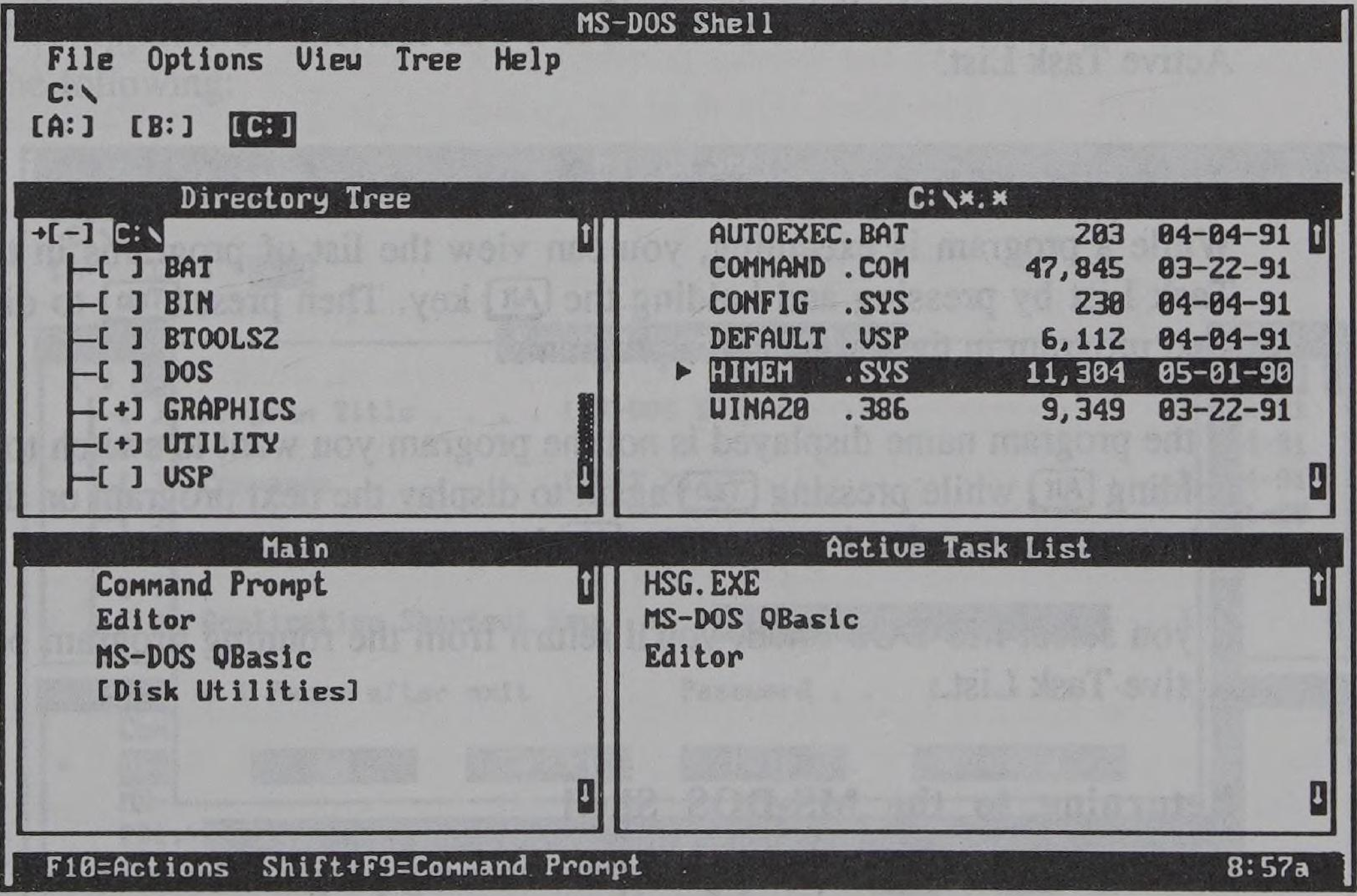
Also, the hard drive should have at least several megabytes free. If you have a couple of large programs running (such as word processors, graphics programs or databases), each data block to be stored could be up to 640K in size.

If there isn't enough free space on your hard drive, the Task Swapper beeps to let you know that it cannot store this information.

The list of active programs

To use the Task Swapper, first you must enable it. Select the **Options** menu either with the mouse or by pressing Alt followed by O. Then select **Enable Task Swapper.**

A fourth area will open at the lower right section of the screen called Active Task List.



```
                              MS-DOS Shell
  File  Options  View  Tree  Help
  C:\
 [A:]  [B:]  [C:]
            Directory Tree                          C:\*.*
 +[-] C:\                            AUTOEXEC.BAT        203  04-04-91
   ├─[ ] BAT                          COMMAND .COM     47,845  03-22-91
   ├─[ ] BIN                          CONFIG  .SYS        230  04-04-91
   ├─[ ] BTOOLSZ                      DEFAULT .VSP      6,112  04-04-91
   ├─[ ] DOS                        ► HIMEM   .SYS     11,304  05-01-90
   ├─[+] GRAPHICS                     WINA20  .386      9,349  03-22-91
   ├─[+] UTILITY
   └─[ ] VSP
                  Main                            Active Task List
        Command Prompt                    HSG.EXE
        Editor                            MS-DOS QBasic
        MS-DOS QBasic                     Editor
        [Disk Utilities]

  F10=Actions  Shift+F9=Command Prompt                                 8:57a
```

The MS-DOS Shell with Task Swapper enabled

When the Task Swapper is switched on, the program list area is vertically divided into two areas. The area on the left displays the list of programs in the current group and the area on the right displays the list of active (stored) programs. When first activated, the Active Task List will be empty.

Switching from one program to the next

When Task Swapper is enabled, each program started from the program list or file list areas is automatically entered in the list of active programs. To switch between programs, press Alt + Esc.

If only one program has been started, pressing Alt + Esc will return you to the MS-DOS Shell without quitting the program.

Any other program can be started from the MS-DOS Shell. Pressing Alt + Esc always takes you to the next program in the list of active programs. To return immediately to the MS-DOS Shell, without exiting any running programs, press Ctrl + Esc.

Another method for adding programs to the Active Task List is to hold down the Shift key while double-clicking on the program name with the left mouse button. Alternatively, use the cursor keys to select the program, then hold down the Shift key while pressing Enter to select the program.

This enters the name on the Active Task List. However, the program is not executed immediately. To load and run any program on the Active Task List, select the program from the list and press Enter. Press Tab if necessary to highlight the Active Task List.

An easier way to switch between programs

While a program is executing, you can view the list of programs in the Active Task List by pressing and holding the Alt key. Then press Tab to display the next program in the list of active programs.

If the program name displayed is not the program you want to switch to, continue holding Alt while pressing Tab again to display the next program on the list. To select a program, simply release the Alt key.

If you select MS-DOS Shell, you'll return from the running program back to the Active Task List.

Returning to the MS-DOS Shell

Pressing Ctrl + Esc takes you from a running program back to the MS-DOS Shell. You can activate another program from the list of active programs any time you want.

Using key combinations to switch between programs

Application shortcut keys can be assigned for even easier switching. The shortcut keys assigned must use the key combination of Ctrl + letter key, Shift + letter key or Alt + letter key.

Also, the program must have been assigned to the program list area as a program item either at the Main area or as part of a program group.

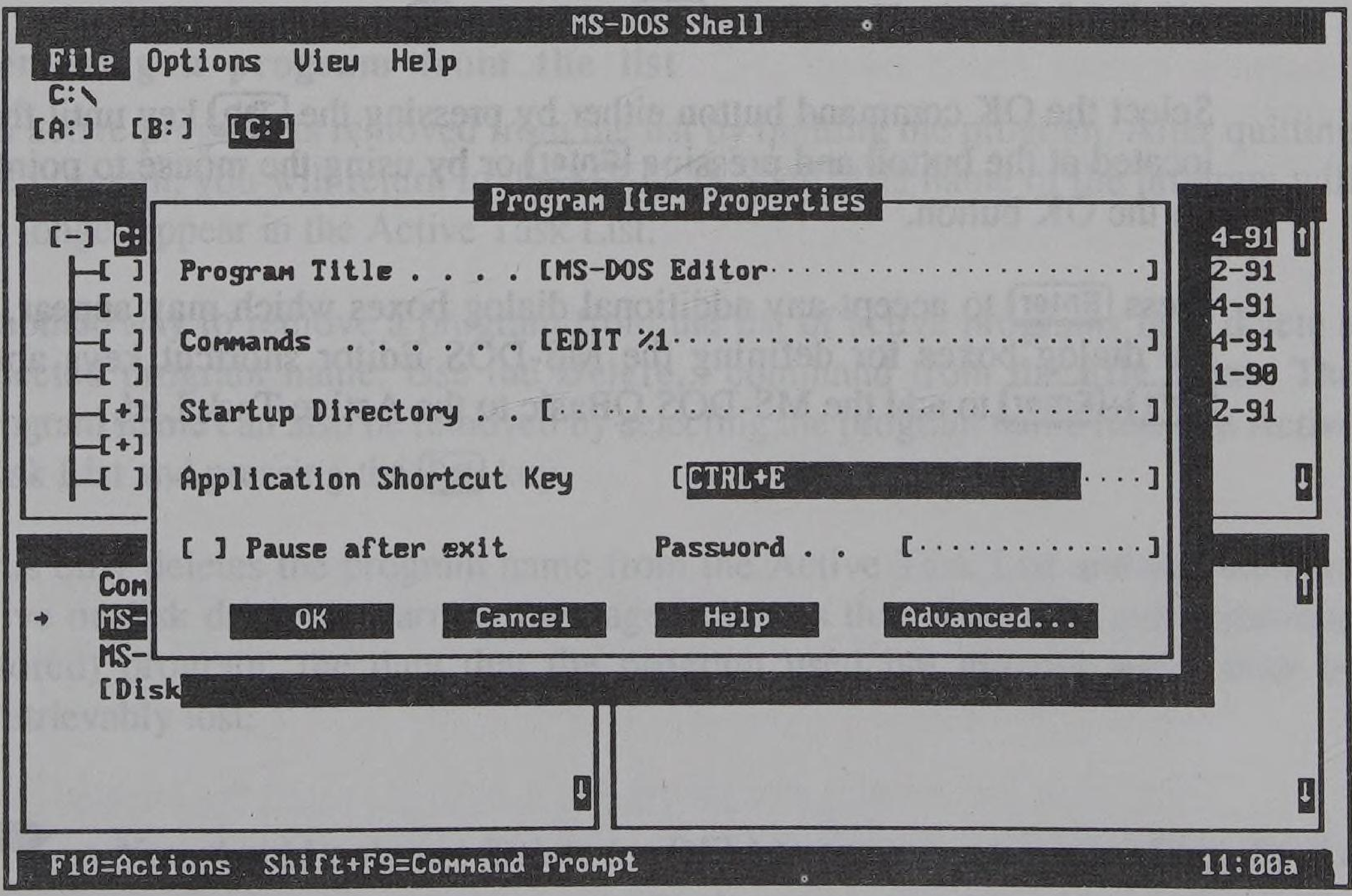
Specifying an application shortcut key

For example, we want to define Ctrl+E the shortcut key for the MS-DOS Editor (called Editor in the program list area). To define an application shortcut key for the MS-DOS Editor, switch to the program list area and select Editor.

Next, activate the **File** menu and select the **Properties...** command. This opens the Program Item Properties dialog box. It displays the name, the command to execute the program (with any optional parameters), the startup directory and any defined application shortcut key and password.

Press Tab (or use the mouse pointer) to move the cursor to the Application Shortcut Key text box. Specify the desired key combination by pressing and holding Ctrl or Shift or Alt while pressing a letter key. In this example, we're using Ctrl+E as the shortcut key. So, hold down Ctrl and press E.

The Application Shortcut Key text box should now display CTRL+E as shown in the following:



Program Title . . . . [MS-DOS Editor]

Commands . . . . . . [EDIT %1]

Startup Directory . . [ ]

Application Shortcut Key [CTRL+E]

[ ] Pause after exit

Password . . [ ]

The Program Item Properties dialog box

Select the OK command button either by pressing the Tab key until the cursor is located at the button and pressing Enter or by using the mouse to point and click on the OK button.

Notice in the Commands text box that EDIT %1 is listed. If a % (indicating a parameter) is listed in the Commands text box, a second Program Item Properties dialog box is displayed requesting information for the parameter.

The MS-DOS Editor will prompt the user to enter a filename every time the program is started because of the %1 parameter in the command line. We'll discuss installing programs with parameters later in this section.

Since we're only interested in defining shortcut keys here, press Enter to accept the second Program Item Properties dialog box.

Press Shift+Enter to add the MS-DOS Editor to the Active Task List. If the File to Edit dialog box appears, press Enter. Again, at this time we're only interested in defining the shortcut keys for the MS-DOS Editor.

Next we want to define Ctrl+Q as a shortcut key for MS-DOS QBasic. Switch to the program list area and use the cursor keys to select MS-DOS QBasic.

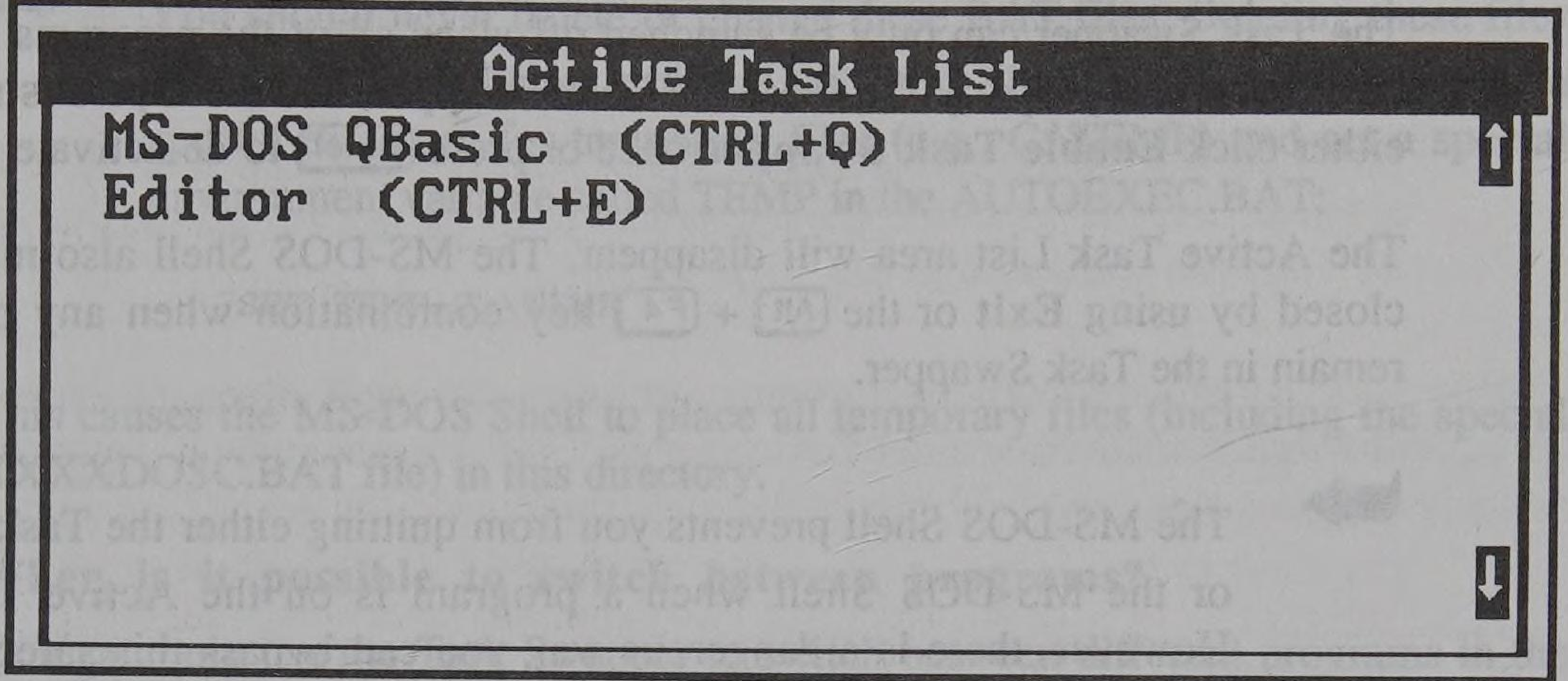
Next, activate the **File** menu and select the **Properties...** command. As with the MS-DOS Editor example, the Program Item Properties dialog box opens.

Press Tab (or use the mouse pointer) to move the cursor to the Application Shortcut Key text box. We want to define Ctrl+Q as the shortcut key for MS-DOS QBasic. Hold down Ctrl and press Q.

Select the OK command button either by pressing the Tab key until the cursor is located at the button and pressing Enter or by using the mouse to point and click on the OK button.

Press Enter to accept any additional dialog boxes which may appear, similar to the dialog boxes for defining the MS-DOS Editor shortcut keys above. Press Shift+Enter to add the MS-DOS QBasic to the Active Task List.

Now the Active Task List should appear like this:



Active Task List

MS-DOS QBasic (CTRL+Q)
Editor (CTRL+E)

The shortcut keys appear with the program name in the Active Task List

Reserve Shortcut Keys

The following key combinations are not available:

Ctrl+[Ctrl+C Ctrl+I
Ctrl+H Ctrl+M Shift+Ctrl+[
Shift+Ctrl+I Shift+Ctrl+H Shift+Ctrl+M
Shift+Ctrl+5 *(keypad)* Ctrl+5 *(keypad)*

Removing a program from the list

An active program is removed from the list by quitting the program. After quitting the program, you will return to the MS-DOS Shell. The name of the program will no longer appear in the Active Task List.

Another way to remove a program from the list of active programs is to delete a selected program name. Use the **Delete...** command from the **File** menu. The program name can also be removed by selecting the program name from the Active Task List and pressing the Del key.

This only deletes the program name from the Active Task List and not the hard drive or disk drive. A warning message indicates that when you quit a running (stored) program, the data that the program used but did not store, may be irretrievably lost.

You should only use Delete (or DEL) to remove a program from the list of active programs when absolutely necessary. This doesn't delete the hidden file, which contains data about the program, from the hard drive.

Quitting Task Swapper

The Task Swapper can only be switched off when all of the programs in the list have been quit properly. To close the Task Swapper, use the **Options** menu and either click **Enable Task Swapper** once or press Enter to de-activate it.

The Active Task List area will disappear. The MS-DOS Shell also may not be closed by using **Exit** or the Alt + F4 key combination when any programs remain in the Task Swapper.

The MS-DOS Shell prevents you from quitting either the Task Swapper or the MS-DOS Shell when a program is on the Active Task List. However, there is a dangerous way you can bypass this safety feature: Simply switching off your computer.

Suppose that you used Shift + F9 or selected Command Prompt from the program list area to leave the MS-DOS Shell temporarily. After executing commands from the MS-DOS command line you could easily forget that the MS-DOS Shell, with its stored files, is still running.

You might even switch off your computer. Now those hidden storage files, which occupy an enormous amount of memory, are still on your hard drive. Before switching off your computer, enter the EXIT command to check whether the MS-DOS Shell is still running.

15.3.5 Special features for more than one program

When you use the Task Swapper, there are a number of special features to consider. The following is a brief list of some simple rules that will help you avoid serious problems.

The meaning of the XXXXDOSC.BAT file

When you start a program from the MS-DOS Shell, a file called XXXXDOSC.BAT is automatically created.

Each of the four "X"s represents a hexadecimal number (0-F). This number is used so that there aren't any files with the same filenames. An example filename might read 8435DOSC.BAT.

This batch file contains everything that is needed to start the selected program. For example, this file contains the line that you entered when defining a new program as a command line. After you exit the program correctly, the XXXXDOSC.BAT file is automatically deleted.

You should never delete or change these BAT files. Deleting these files could lead to problems with the MS-DOS Shell. It's also best to create a special directory for temporary files (e.g., C:\TEMP) and set a special environment variable called TEMP in the AUTOEXEC.BAT:

```
SET TEMP=C:\TEMP
```

This causes the MS-DOS Shell to place all temporary files (including the special XXXXDOSC.BAT file) in this directory.

When is it possible to switch between programs?

You can only use the Task Swapper to switch between different programs in the Active Task List after the XXXXDOSC.BAT file is finished running. For example, if the command line for a program is:

```
WORD TEXT1.DOC | COPY TEXT1.DOC A: | DIR
```

you cannot switch while editing in Word. While you are using Word, the batch file hasn't finished running.

Use caution when deleting files

When using the Task Swapper, be especially careful when deleting files. The MS-DOS Shell has to place parts of memory in temporary storage on the hard drive by creating special files. These files use the .TMP extension and begin with the character sequence "~WO".

While you are working with the MS-DOS Shell, do not delete, change or rename these files. If you do make changes to these files, you won't be able to switch between programs. Also, your computer will probably crash and all of the unsaved data from the applications currently running will be lost.

There has to be room to switch

In order to use the Task Swapping option, there must be sufficient free space on your hard drive. If there isn't enough room to store programs and their data on the hard drive, you won't be able to switch.

The speed required for switching between programs can be increased considerably by using a faster hard drive. Another option is to optimize your system for speed by using the available utilities provided with MS-DOS 5.0, such as SMARTDRV.SYS and FASTOPEN.EXE.

Updating the contents of the MS-DOS Shell

If you used a program and changed the contents of the drive and then returned to the MS-DOS Shell, you'll notice that the Directory Tree/file list area hasn't recorded the changed data. This is normal for all changes to files and directories that are caused by programs run from the program list area.

To update the display of the current status of all of the files in a drive, double-click the drive icon or select the drive icon and press Enter. An alternative is to select the **Refresh** command in the **View** menu.

Use caution with programs that note the data location from storage media

When working and switching between several programs, be careful when selecting programs to run.

All programs that use the location of data on a storage media (floppy diskette or hard drive) are forbidden. So, you cannot start a program for optimizing and clearing the hard drive from the MS-DOS Shell and then switch to the MS-DOS Shell or another program. When you want to work on the system, leave the MS-DOS Shell (Alt + F4) and start the program from the command line.

Since commands such as FASTOPEN and programs such as cache programs assume that they are running alone in the PC, they use information about the absolute storage locations. Therefore, these types of programs cannot run from the MS-DOS Shell. You must install them before starting the MS-DOS Shell for the first time.

15.3.6 Installing and changing programs

One important feature of the Program list is that the program list area can be customized. This makes it easier for you to run applications.

Adding a program item

To add a new program item, first switch to the program list area and then activate **New...** from the **File** menu. A dialog box will ask whether you want to set up a new program (Program Item, which is the default setting) or a new group (Program Group).

After pressing Enter or clicking OK, the Add Program dialog box appears. This dialog box lets you enter important program information.

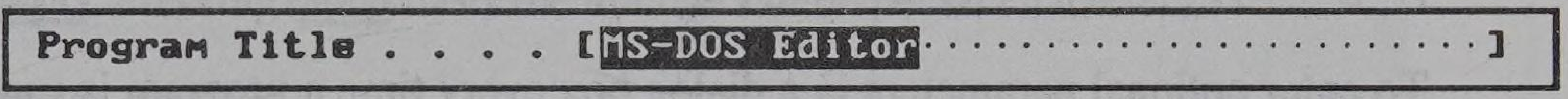
The two dialog boxes for setting up a program are also used for changing a program item.

Changing a program item

Before a program item can be changed, it must be selected with either the mouse or the cursor keys. The properties of the selected program item can be changed by activating **Properties...** from the **File** menu. This command opens the Program Item Properties dialog box, in which you can enter the program definitions.

Program Title

In the Program Title text box, type the name of the program as it should be displayed in the program list. This name will serve as the identifier for starting the program. The title can contain up to 23 characters. Select a clear and logical name. To move to the next text box, press Tab.



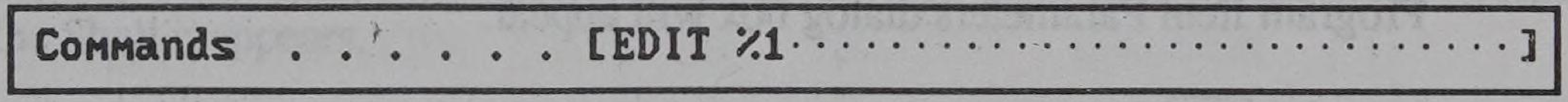
```
Program Title . . . . [MS-DOS Editor·······················]
```

The Program Title text box from the Program Item Properties dialog box

Commands

The next text box is labeled Commands. Enter the command that normally would be entered at the system prompt on the MS-DOS command line to run the program.

For example, if you have the program Word installed in the directory C:\WORD and used PATH to define a path to this directory, you only need to enter WORD. If the path isn't defined, you must enter the complete command, including the path: C:\WORD\WORD. Up to 255 characters can be used.



```
Commands . . . . . . . [EDIT %1·····························]
```

The Commands text box from the Program Item Properties dialog box

Additional commands can be specified, batch files can be run and replaceable parameters can be included as part of the command line.

The MS-DOS Shell executes each command in the order it appears in the Commands text box. If more than one command is entered, the additional commands must be separated by a semicolon (;). Each semicolon must have one or more spaces on each side of it.

Batch files can be run by including the CALL command in the command line. For example, to run a batch file named INIT.BAT prior to running a program and a batch file called CLEANUP.BAT after the program ends, type the following command line:

```
CALL INIT ; PROGRAM ; CLEANUP
```

The commands must be separated by a space, semicolon and another space. The programs will not run correctly if the spaces are omitted.

Using Parameters

Parameters are additional information given to a program when it is started. For example, when the MS-DOS Editor or MS-DOS QBasic are started from the program list area, a dialog box will request the name of an optional file to load. You can enter a filename here and it will load as soon as the program starts.

This additional information (the filename to load) is passed in a variable. Many programs, such as Word, can use optional parameters in this way.

To enter optional parameters in a dialog box every time a program is run from the program list, enter a replaceable parameter in the Commands text box of the Program Item Parameters dialog box. Replaceable parameters are entered as a percentage sign (%) followed by a number from 1 through 9.

In the previous illustration, the MS-DOS Editor will prompt the user to enter a filename every time the program is started because of the command line:

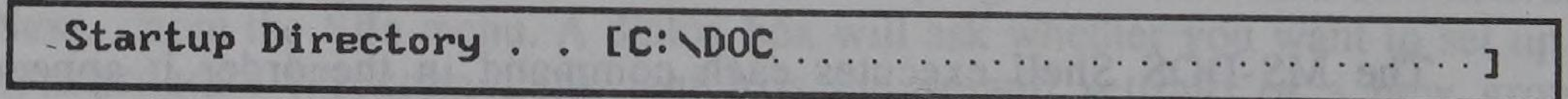
```
EDIT %1
```

Including Replaceable Parameters

To include a replaceable parameter with a program item, first select the item from the program list area, then open the **File** menu and choose **Properties....** The Program Item Parameters dialog box will appear.

Use the Tab key or the mouse to activate the Commands text box. Enter the command to execute the program, followed by the necessary replaceable parameters. Enter each parameter as %1, %2, %3, etc., and select the OK button.

Startup Directory

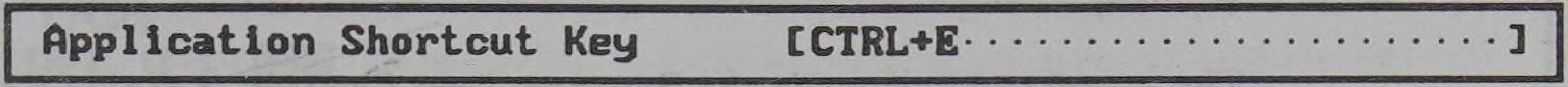


```
Startup Directory . . [C:\DOC.................................]
```

The Startup Directory text box from the Program Item Properties dialog box

In the Startup Directory text box, you can specify which directory becomes the current directory before the program starts. For example, if you store all of your word processing documents created by Word in the C:\DOC directory, enter this directory here. When Word is run from the program list area, C:\DOC will become the current directory.

Application Shortcut Key



```
Application Shortcut Key      [CTRL+E·····················]
```

Application shortcut key text box in the Program Item Properties dialog box

This text box lets you combine Ctrl, Alt, Shift and another key (e.g., a letter) to a program. This assignment allows you to change directly from the Task Swapper to the program.

To enter the key code in this text box, hold down the Ctrl, Alt or Shift key and press a letter, number or function key. The MS-DOS Shell then enters the key combination pressed in this text box. (You cannot use the numbers 0 - 9 with Shift. Pressing Shift turns the numbers into special characters.)

Pause after exit

If you used Tab or the mouse to move to this check box, you can press the Spacebar or click on Pause after exit to place an [X] inside the brackets, which switches this feature on or off.

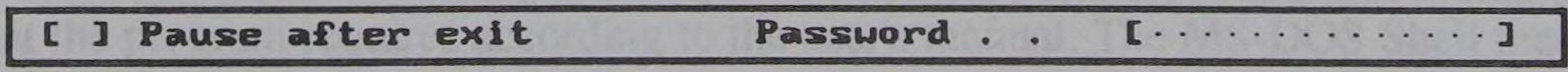
The X signifies that the MS-DOS Shell will wait for you to press any key before returning to the MS-DOS Shell screen after exiting the program. If there is no X, the feature is switched off and the MS-DOS Shell screen reappears immediately after you leave a program.

You may not want to clear the screen and open the MS-DOS Shell immediately after exiting a program because some programs (like the MS-DOS commands DIR, CHKDSK and MEM) display information on the screen as they end. Calling the MS-DOS Shell immediately will prevent you from reading the screen before the Shell reappears.

Password

If you enter a password, containing one to eight characters, in the Password text box before the program is started, you will be prompted for a password. The keystrokes entered for a password are not displayed on the screen.

You cannot make changes, copy or delete this program item until the correct password is entered. This option of the MS-DOS Shell is case sensitive, which means that it recognizes the difference between upper and lowercase characters.



```
[ ] Pause after exit          Password . .     [···············]
```

Pause after exit and Password items of the Program Item Properties dialog box

OK

Pressing OK (use Tab or the mouse to select it and confirm by pressing Enter or clicking the left mouse button on it) accepts the settings you've made as correct.

OK
Cancel
Help
Advanced...

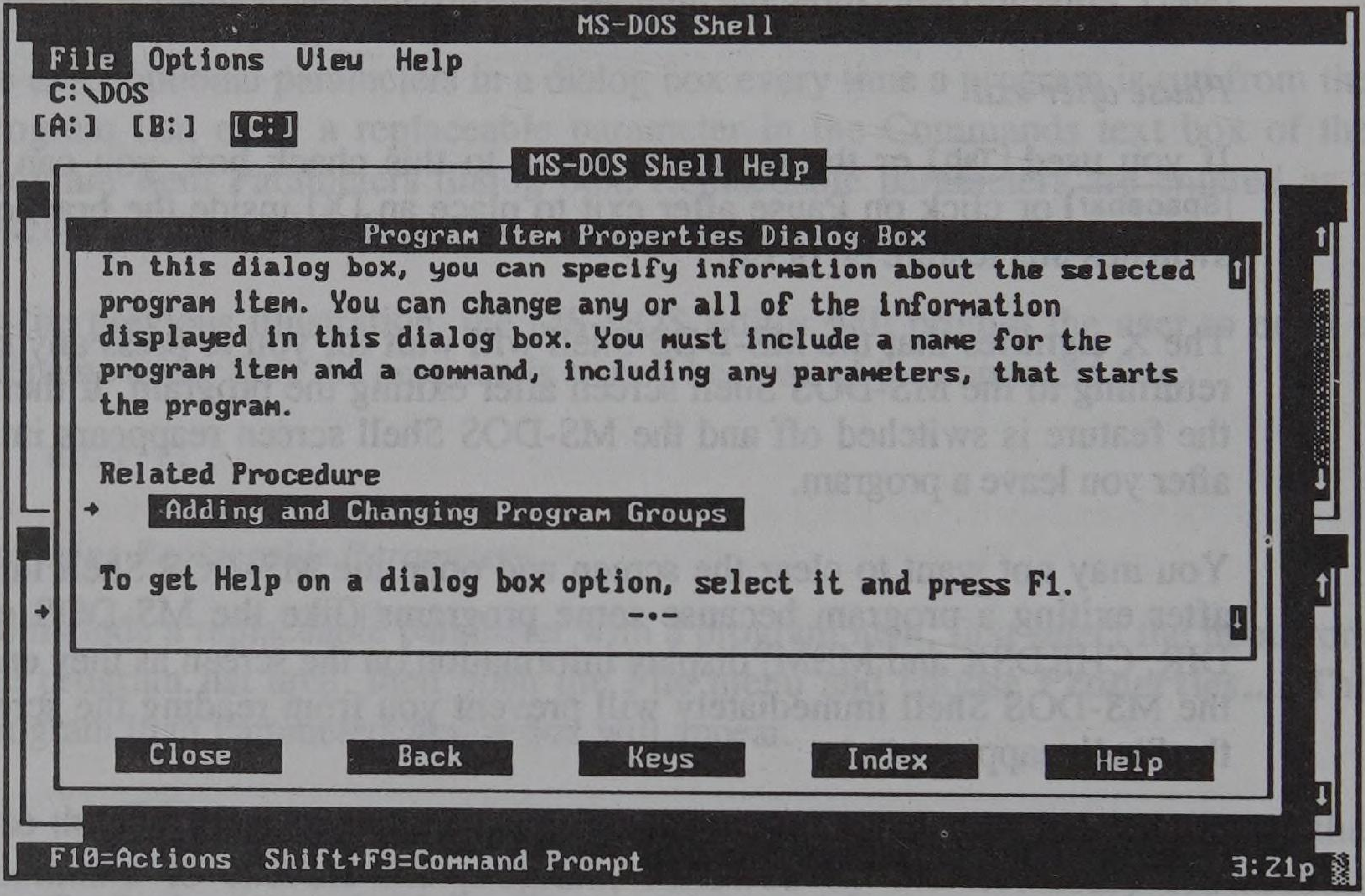
The command buttons in the Program Item Properties dialog box

Cancel

Pressing Cancel will cancel all of the settings and all of the previously entered data in this dialog box will be lost.

Help

Selecting Help displays the MS-DOS Shell help screen for the Program Item Properties dialog box.


MS-DOS Shell
File
Options
View
Help
C:\DOS
[A:]
[B:]
[C:]
MS-DOS Shell Help

Program Item Properties Dialog Box

In this dialog box, you can specify information about the selected program item. You can change any or all of the information displayed in this dialog box. You must include a name for the program item and a command, including any parameters, that starts the program.

Related Procedure

→ Adding and Changing Program Groups

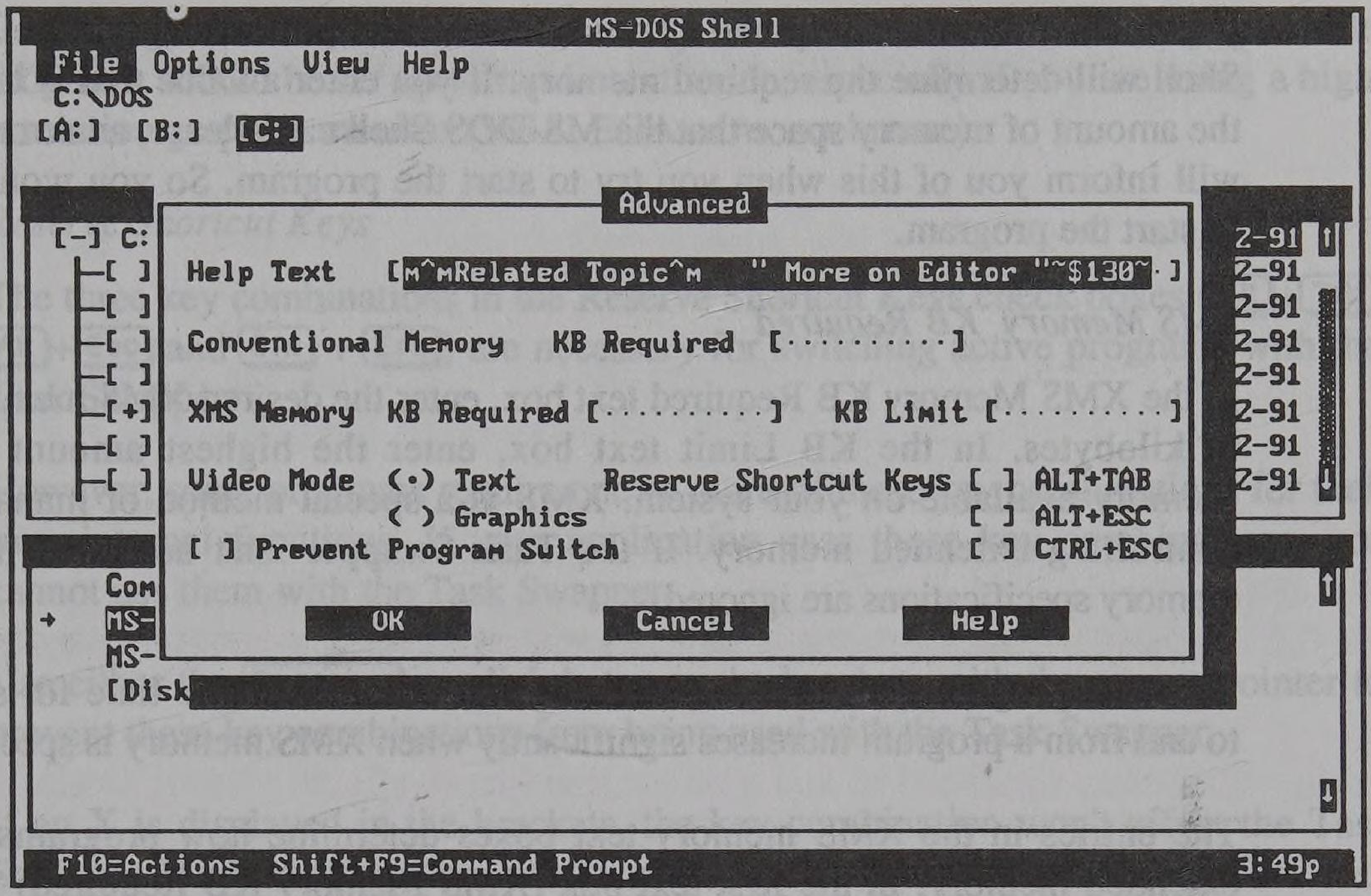
To get Help on a dialog box option, select it and press F1.

Close
Back
Keys
Index
Help
F10=Actions
Shift+F9=Command Prompt
3:21p

The Program Item Properties help screen

Advanced...

Selecting the Advanced... button displays a new dialog box where you can make advanced settings.



```
Help Text    [M^MRelated Topic^M    " More on Editor "~$130~·]

Conventional Memory   KB Required [··········]

XMS Memory  KB Required [··········]  KB Limit [··········]

Video Mode  (•) Text      Reserve Shortcut Keys [ ] ALT+TAB
            ( ) Graphics                        [ ] ALT+ESC
[ ] Prevent Program Switch                      [ ] CTRL+ESC
```

The Advanced Mode dialog box

Help Text

The Help Text text box lets you define a string of explanatory text that will appear when you select a program and call the **Help** menu by pressing F1. Up to 478 characters can be entered and word wrapping is automatic.

To control word wrapping, enter the characters "^M" anywhere a new line should begin. Enter blank lines by typing "^M" twice (^M^M).

The caret "^" is usually available as a key on your keyboard, if it is not, press Alt + 9 4.

Remember that you must activate Num Lock to enter characters from the numeric keypad.

Conventional Memory KB Required

In the Conventional Memory KB Required text box, enter the amount of memory that the program requires according to the user's manual. The MS-DOS Shell then makes this memory area available.

Example

The word processing program Word has a 638K EXE file. The MS-DOS Shell cannot release that much memory. According to the Word manual, the required (minimum) memory space amounts to 384K. This is the value that must be entered.

If you don't know the required amount, leave this text box empty. The MS-DOS Shell will determine the required memory. If you enter a value that is larger than the amount of memory space that the MS-DOS Shell can release, an error message will inform you of this when you try to start the program. So you won't be able to start the program.

XMS Memory KB Required

In the XMS Memory KB Required text box, enter the desired XMS memory space in kilobytes. In the KB Limit text box, enter the highest amount of XMS Memory available on your system. XMS is a special method of managing and controlling extended memory. If the Task Swapper isn't activated, the XMS memory specifications are ignored.

For most programs it's best to leave this text box blank. The time for switching to and from a program increases significantly when XMS memory is specified.

The entries in the XMS memory text boxes determine how programs will use extended memory. In the first text box (XMS Memory KB Required) you must determine the minimum amount of XMS memory that must be available in order to start a program.

In most cases, you should enter 0 so that the presence of extended memory doesn't influence starting the program. If a program requires a specific amount of XMS memory to run properly, enter this amount in the KB Required text box.

Cache programs, which require a certain amount of XMS memory for temporarily storing data, are an example. If a cache program uses conventional memory instead of XMS memory, starting the program could be disastrous because conventional memory is suddenly decreased drastically. Use this text box to prevent a program from starting with conventional memory.

✎ Entering any value other than 0 delays the start process for the program.

The KB Limit text box enables you to set the maximum amount of XMS memory that can be used for the program. This option is useful for programs that always use all of extended memory. When working with more than one program (with the Task Swapper), usually you shouldn't use extended memory. You can prevent a program from using XMS memory by entering a zero in this text box.

Video Mode

The Video Mode options are used to determine whether the program runs in text mode or graphics mode. To select graphics mode either click or use the Spacebar to activate the Graphics option. If Text is selected and you switch to graphics mode while the program is running, the Task Swapper might not switch. You must also activate Graphics for programs that only run in graphics mode.

Graphics mode requires more memory than text mode. Whenever possible, programs should be set up to run in text mode, especially if you are using a high-resolution graphics monitor (VGA, EGA or monochrome).

Reserve Shortcut Keys

The three key combinations in the Reserve Shortcut Keys check boxes (Alt+Tab, Alt+Esc and Ctrl + Esc) are necessary for switching active programs with the Task Swapper.

However, some programs require one or more of these key combinations for their own internal functions. If your application uses these key combinations, you cannot use them with the Task Swapper.

Use either the Spacebar or click between the brackets with the mouse pointer to prevent these key combinations from being used with the Task Swapper.

If an X is displayed in the brackets, the key combination won't affect the Task Swapper. If all three key combinations are selected, the program cannot work with the Task Swapper.

Prevent Program Switch

Selecting the Prevent Program Switch check box and placing an X in the brackets prevents switching from this program. To exit the program you must return to the MS-DOS Shell.

OK

Selecting OK (use Tab to go to OK, then press Enter or click with the mouse) accepts the settings as correct.

Cancel

Selecting Cancel ignores all of the settings in the dialog box. Any entries previously made are lost.

Help

Calls the appropriate pages of the MS-DOS Shell help screen.

Tips for creating a new program

When adding a new program to the program list area, you may want to enter one or more MS-DOS commands. First use the Tab key to move to the program list area. Then select **New...** from the **File** menu. Press Enter to accept the default, Program Item.

The Add Program dialog box appears. Type the name of the program in the Program Title text box and press Enter.

Then type the command in the Commands text box. If you enter more than one command, be sure that you separate each command with a space, semicolon and another space.

If the program is located in a specific directory, you can enter that directory in the Startup Directory text box. Also, if desired, enter a shortcut key in the Application Shortcut Key text box and a password in the Password text box.

If an X doesn't appear in the Pause after exit check box, the command will return automatically to MS-DOS Shell. Otherwise, the command will wait until you press a key to return to MS-DOS Shell.

Then use the cursor keys to move the selection cursor to the program name. Press Enter. The program then executes the commands you entered in the Commands text box.

When the command has executed, the message:

```
Press any key to return to MS-DOS Shell . . . .
```

will appear if the Pause after exit check box was activated.

The command line can be used to start batch files as well as COM or EXE programs. Since batch files can be almost any length, the limitations in the length of the command lines in the MS-DOS Shell no longer apply.

For example, a file called WORK.BAT could be created in the DOS directory and called from the command line. However, a batch file cannot be called directly in an MS-DOS Shell command line. In other words, you cannot type:

```
WORK.BAT
```

Your batch file will end at the completion of WORK.BAT and none of the remaining command elements in the command line will be executed. Instead, use CALL. For example, to call the WORK.BAT batch file for the example above, use the following command line:

```
CALL WORK.BAT
```

The commands in program lines in the MS-DOS Shell work like batch files and are subject to the same rules.

You can also use other elements of batch programming in command lines of the MS-DOS Shell (e.g., IF). The batch file:

```
IF EXIST MS-DOS Shell.BAT TYPE MS-DOS Shell.BAT
ECHO End of program
PAUSE
```

corresponds to the following command line in the MS-DOS Shell:

```
IF EXIST MS-DOS Shell.BAT TYPE MS-DOS Shell.BAT ; ECHO End of
program ; PAUSE
```

The commands must be separated by a space, semicolon and another space. The programs will not run correctly if the spaces are omitted.

15.3.7 Installing programs with parameters

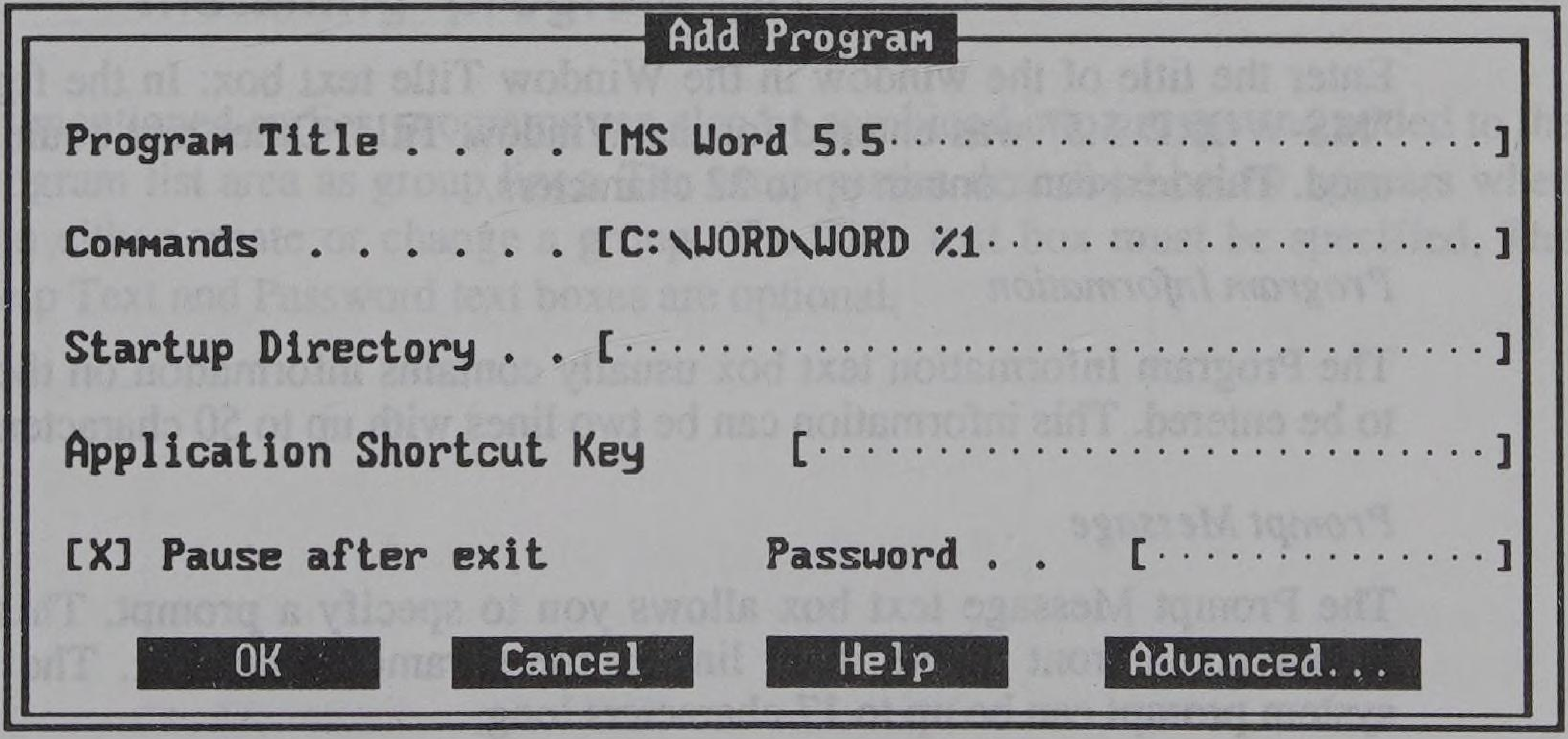
If a program requires one or more parameters to run, a dialog box for entering the proper parameters appears on the screen after you select the program item. The MS-DOS Editor and MS-DOS QBasic, both of which are part of MS-DOS 5.0 and installed in the MS-DOS Shell, have these dialog boxes.

You can specify the file to edit in the dialog box of the editor and specify which BASIC file to read in the dialog box of QBasic.

You can install a word processing program so that when you call the program, you can specify, in a dialog box, the file that should be loaded with the word processing program.

Enter "%1" ("%2", "%3", through "%9" are also valid) to start a program with one or more parameters. Enter "%1" after the program name in the Commands text box of the Program Item Properties dialog box.

For example, to start Word with a dialog box for parameters, enter "WORD %1" or "C:\WORD\WORD %1" after "Commands".



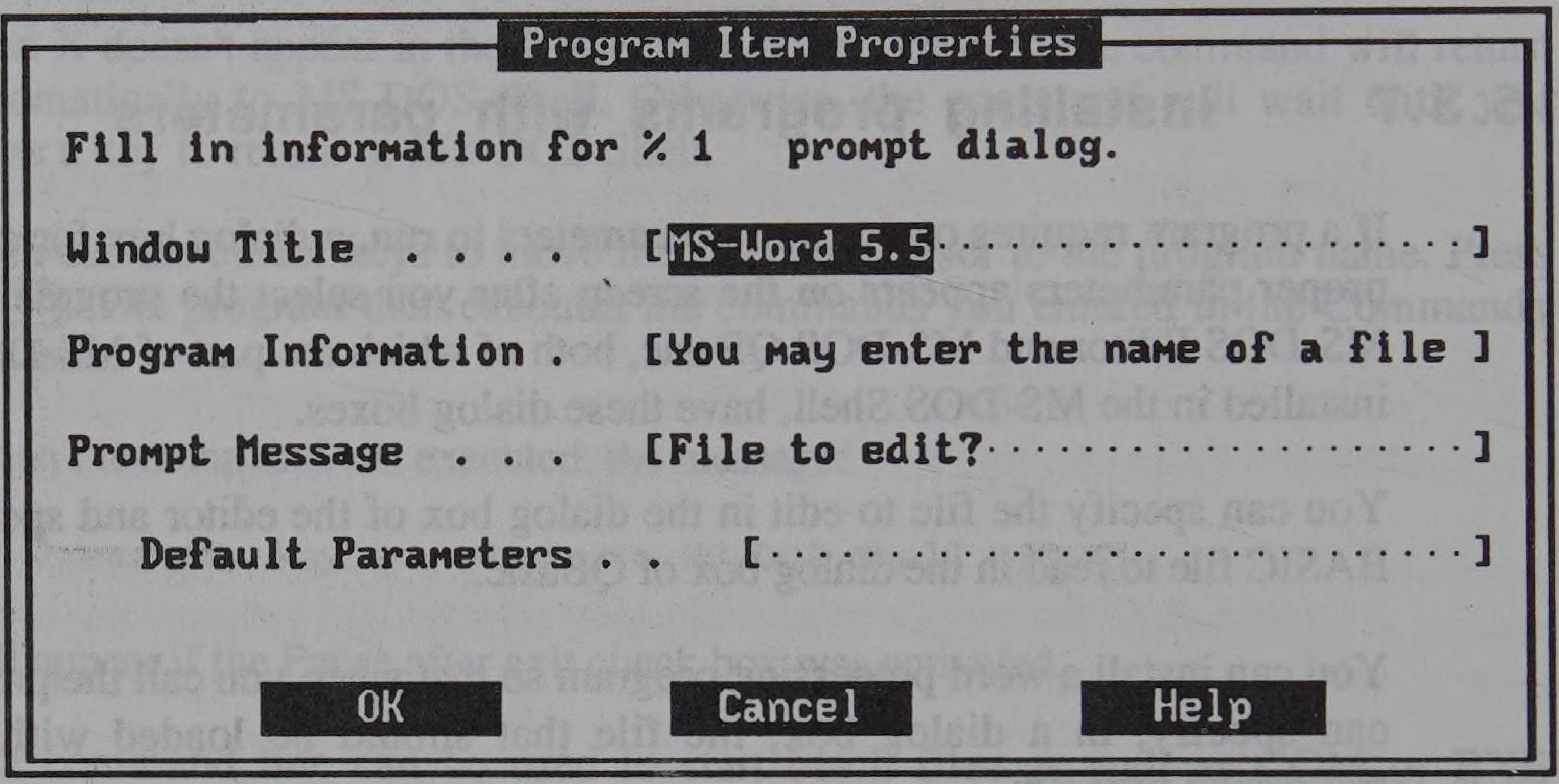
```
                        Add Program

Program Title . . . . [MS Word 5.5·························]

Commands  . . . . . . [C:\WORD\WORD %1·····················]

Startup Directory . . [····································]

Application Shortcut Key      [····························]

[X] Pause after exit          Password . .   [·············]
```

Installing Word 5.5 to start with a dialog box for entering a filename

The parameter window

For each variable (%n) entered in the command line, the MS-DOS Shell displays a parameter window when the program item is selected.

When creating the program item, enter information in the dialog box to customize the dialog box which will be displayed for your program item.



```
                     Program Item Properties

  Fill in information for % 1   prompt dialog.

  Window Title  . . . .     [MS-Word 5.5·····················]

  Program Information .     [You may enter the name of a file ]

  Prompt Message  . . .     [File to edit?···················]

     Default Parameters . .   [······························]
```

The dialog box for entering information about additional parameters

When entering a "%1" after a command in the Commands text box when creating or changing a command line and selecting OK, you'll see a dialog box like the one above. It lets you add additional information about the parameter.

The top line indicates with which parameter this dialog box is associated. This is useful when multiple parameters are passed to a program item.

Enter the title of the window in the Window Title text box. In the figure above, "MS-WORD 5.5" was entered for the Window Title. Other text could have been used. This text can contain up to 22 characters.

Program Information

The Program Information text box usually contains information on the parameter to be entered. This information can be two lines with up to 50 characters each.

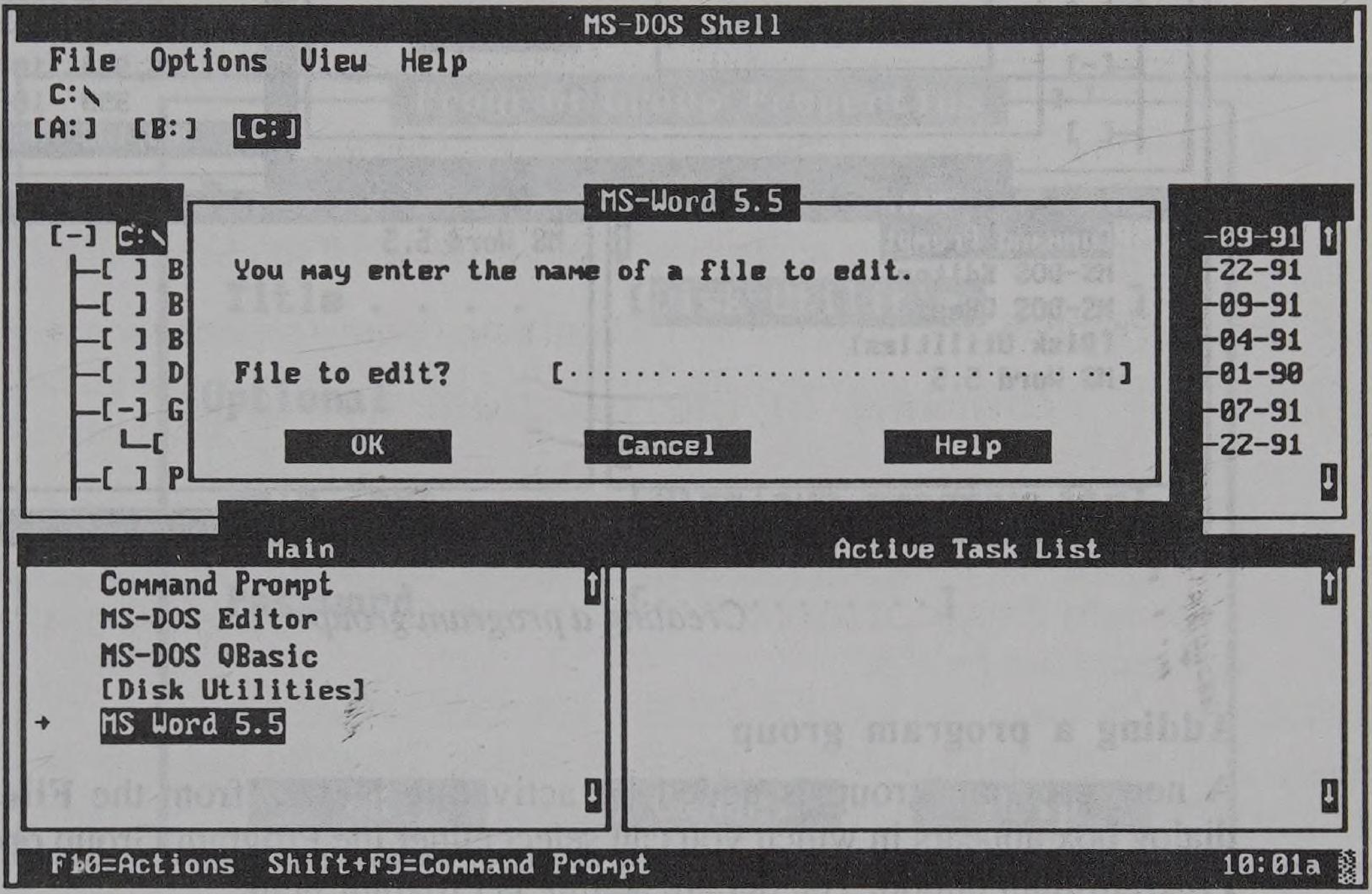
Prompt Message

The Prompt Message text box allows you to specify a prompt. This prompt is displayed in front of the input line in the parameter window. The text of the system prompt can be up to 17 characters long.

Default Parameters

The Default Parameters text box can contain a default parameter which is displayed in the input line when the parameter window appears. This default parameter can be accepted by pressing (Enter) or can be edited. The text of the default parameter can be up to 19 characters long.

Either click OK or press (Enter) to accept the entries. Select Cancel if you don't want to accept the entries. If you specified more than one parameter in the Command text box, the setting options for the next parameter window appear.


MS-DOS Shell
File Options View Help
C:\
[A:] [B:] [C:]
MS-Word 5.5
OK
Cancel
Help
Main
Command Prompt
MS-DOS Editor
MS-DOS QBasic
[Disk Utilities]
MS Word 5.5
Active Task List
F10=Actions Shift+F9=Command Prompt
10:01a

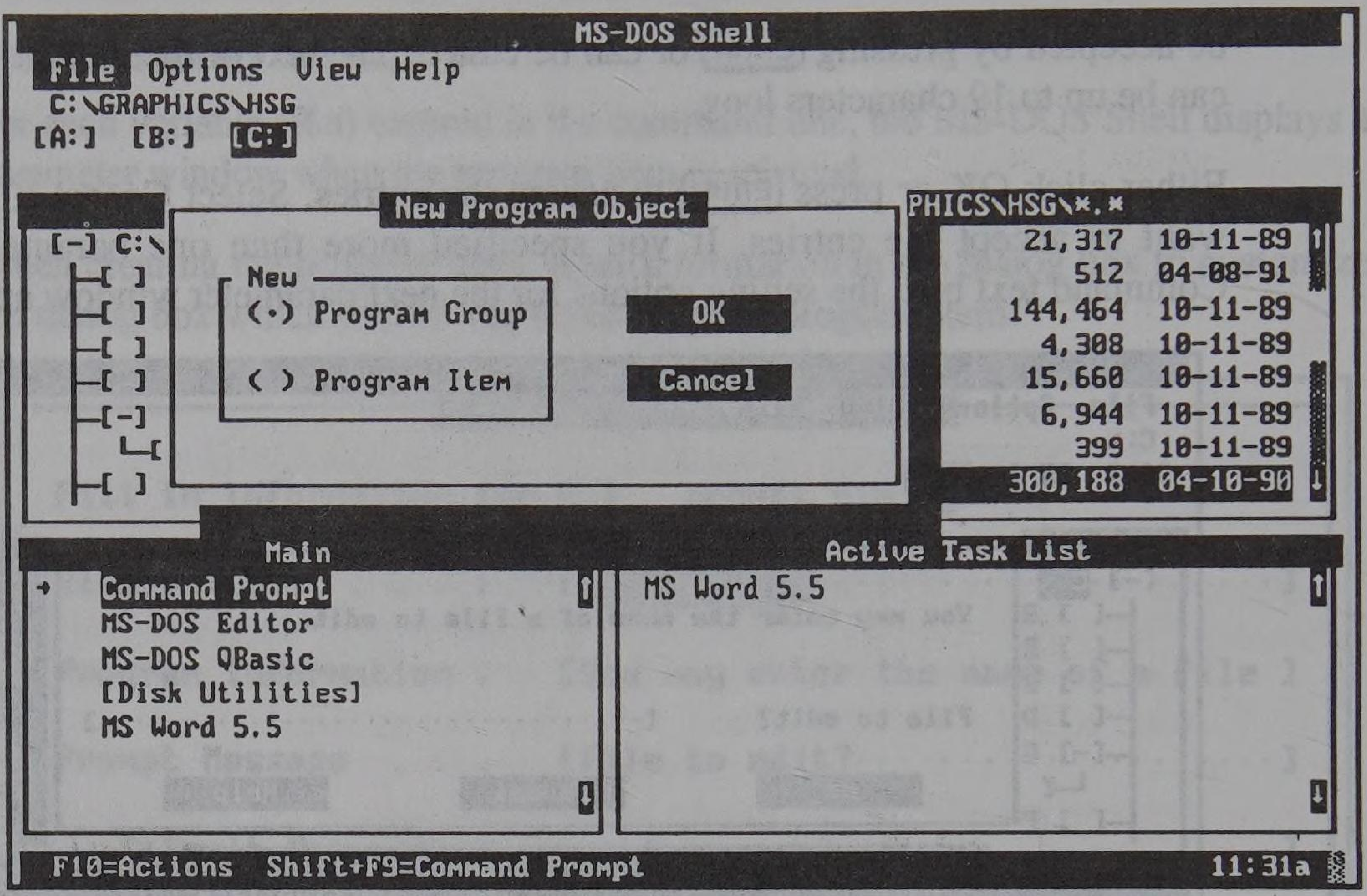
You may enter the name of a file to edit.

File to edit? [.......................................]

Starting MS-Word 5.5 from the Program Item list displays this dialog box

15.3.8 Installing program groups

As mentioned earlier, programs can also be combined into groups and added to the program list area as group lines. The setup screen described below appears when you either create or change a group. The Title text box must be specified, The Help Text and Password text boxes are optional.

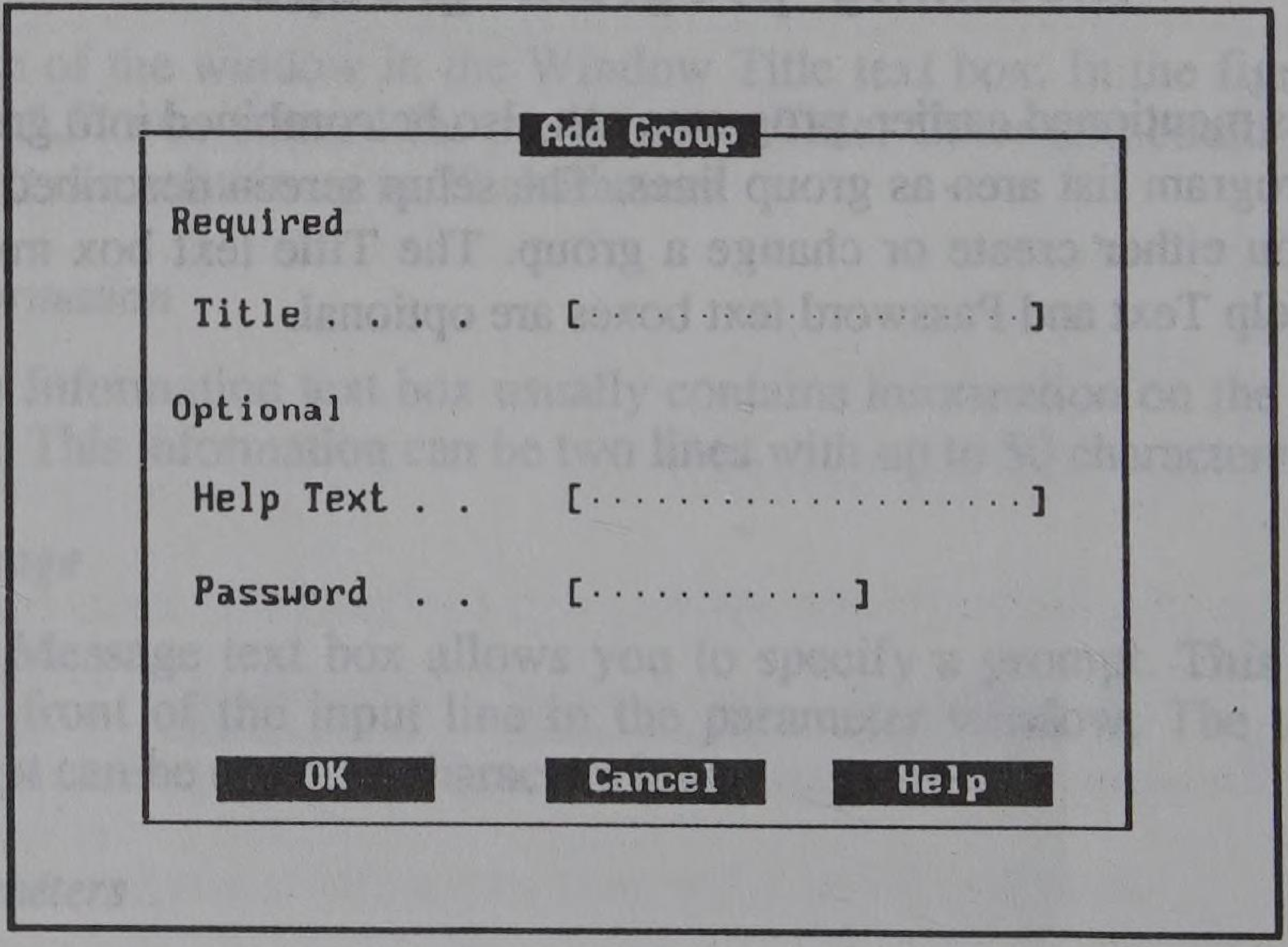

MS-DOS Shell
File
Options
View
Help
C:\GRAPHICS\HSG
[A:]
[B:]
[C:]
[-] C:\
New Program Object
New
(•) Program Group
( ) Program Item
OK
Cancel
PHICS\HSG\*.*
21,317 10-11-89
512 04-08-91
144,464 10-11-89
4,308 10-11-89
15,660 10-11-89
6,944 10-11-89
399 10-11-89
300,188 04-10-90
Main
Command Prompt
MS-DOS Editor
MS-DOS QBasic
[Disk Utilities]
MS Word 5.5
Active Task List
MS Word 5.5
F10=Actions
Shift+F9=Command Prompt
11:31a

Creating a program group

Adding a program group

A new program group is added by activating **New...** from the **File** menu. A dialog box appears in which you can select either the Program Group option or the Program Item option. The default setting is Program Item.

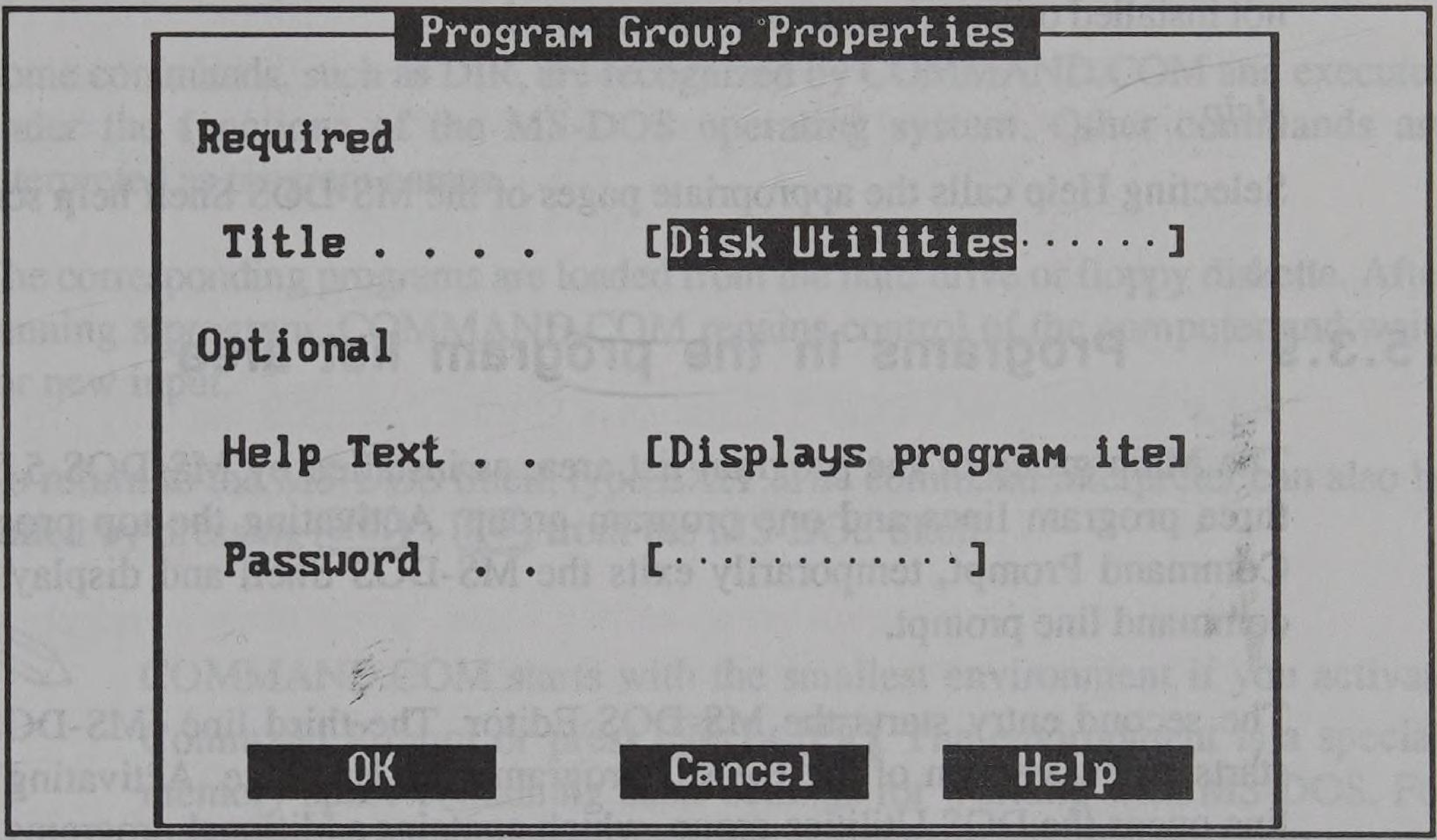
Press the ↑ cursor key or click Program Group with the mouse to select "group". Then either press Enter or click OK. Then the Add Group dialog box appears. You can now define the new program group.


Add Group
Required
Title . . . . [··········]
Optional
Help Text . . [·····················]
Password . . [············]
OK
Cancel
Help

Add Group dialog box

Changing a program group

The following procedure can be used to make changes to an existing program group. First, use the mouse or the cursor key to select the program group to be changed. Then activate **Properties...** from the **File** menu. The Program Group Properties dialog box is displayed as shown below. This dialog box has the same setting options as the Add Group dialog box except that the old values are already in the lines.



Program Group Properties

Required

Title . . . . [Disk Utilities······]

Optional

Help Text . . [Displays program ite]

Password . . [············]

Changing a program group

Title

Enter the text, which will be displayed as the menu entry for the program group, in the Title text box. The title can contain up to 23 characters. For example, to set up a group that contains several word processing programs, the text entry could be "Word processors". Press Tab to move to the next text box, Help Text.

Help Text

The Help Text text box enables you to enter information about the group. Press Tab again to move to the next text box.

Password

By specifying a password you can control who has access to this group. The MS-DOS Shell always asks for this password. When the password is entered, the screen display only shows an asterisk. This is a safety feature that prevents others from seeing your password as you enter it.

The MS-DOS Shell password protection option is case sensitive. In other words, it recognizes the difference between upper and lowercase characters.

Save

Selecting OK (use Tab or the mouse to move to OK, then press Enter or click on it) accepts the settings as entered. A new group has now been created or changes have been made to an existing one.

Cancel

Cancel ignores all of the settings and changes which have been made. The group is not installed or any changes are not accepted.

Help

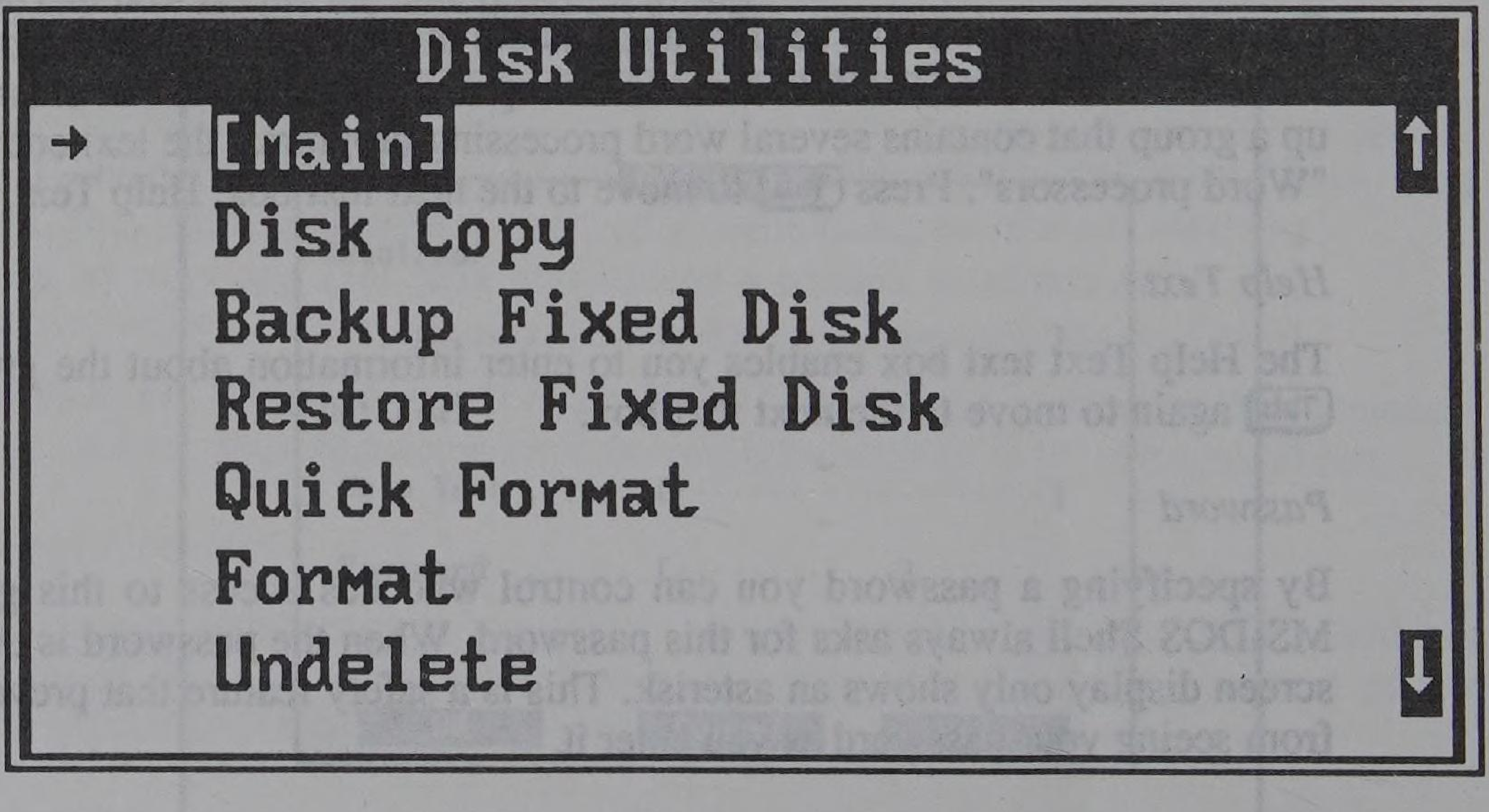
Selecting Help calls the appropriate pages of the MS-DOS Shell help screen.

15.3.9 Programs in the program list area

The Main group of the program list area, as installed by MS-DOS 5.0, contains three program lines and one program group. Activating the top program item, Command Prompt, temporarily exits the MS-DOS Shell and displays the DOS command line prompt.

The second entry starts the MS-DOS Editor. The third line (MS-DOS QBasic) starts a new version of the BASIC programming language. Activating the fourth line opens the DOS Utilities group, which contains additional programs.

The following sections provide an explanation of the program lines as well as the program group.



The Disk Utilities program group

The MS-DOS Command Prompt

The MS-DOS Command Prompt can be used to leave the MS-DOS Shell temporarily and start a new command interpreter (COMMAND.COM).

COMMAND.COM is usually stored in the root directory of the hard drive or diskette from which your computer is booted. COMMAND.COM provides an interface between the user and the computer by taking input from the keyboard and acting on this input.

Some commands, such as DIR, are recognized by COMMAND.COM and executed under the functions of the MS-DOS operating system. Other commands are interpreted as program names.

The corresponding programs are loaded from the hard drive or floppy diskette. After running a program, COMMAND.COM regains control of the computer and waits for new input.

To return to the MS-DOS Shell, type EXIT. The command interpreter can also be called by pressing Shift + F9 from the MS-DOS Shell.

COMMAND.COM starts with the smallest environment if you activate Command Prompt or press Shift + F9. The environment is a special memory space containing basic settings for working with MS-DOS. For example, it contains the settings for the appearance of the system prompt or the current search path.

To call commands or programs from the command line that require a larger environment, use **Properties...** from the **File** menu to change the DOS command prompt program item.

When the Program Item Properties dialog box appears, specify the /E: switch to increase the environment memory after "COMMAND" in the Commands text box. Specify the size for environment memory in bytes following the /E: switch.

The following is an example of the Commands text box setting the environment memory for 512 bytes:

```
COMMAND /E:512
```

This increased environment memory is only available when the command line is activated by using the program item Command Prompt. The key combination Shift + F9 starts COMMAND.COM with the minimum amount of environment memory (160 bytes).

The MS-DOS Editor

Selecting the MS-DOS Editor starts the editor supplied with MS-DOS 5.0. After activating this program item, a dialog box will request the name of the file to be edited.

Refer to Chapter 13 for more information on the MS-DOS Editor.

MS-DOS QBasic

Activating the MS-DOS QBasic program item starts QBasic. After activating this program item, a dialog box will request the name of the file to be edited.

Refer to Chapter 16 for more information on QBasic.

The Disk Utilities

After installing MS-DOS 5.0, the Disk Utilities group is the only group available in the Main group of the program list area. When this item is selected, a new list of commands is displayed. The name "Disk Utilities" appears in the title bar.

The top line is a program group named Main. When Main is selected or Esc is pressed, you will return to the Main group (the top level).

Disk Copy

Disk Copy is used to copy diskettes. First select a source drive and a target drive. The default setting specifies A: as the source drive and B: as the target drive. These settings can be changed. After confirming your choices by pressing Enter, the MS-DOS DISKCOPY command is executed.

Backup Fixed Disk

Use the Backup Fixed Disk program item to make a backup of your hard drive. The default setting is to backup all of the files and subdirectories on drive C:. The setting for the target drive is A:. These settings can be changed.

Restore Fixed Disk

The Restore Fixed Disk program item restores files that were previously backed up with the BACKUP or the Backup Fixed Disk commands. This program uses the RESTORE command. The default setting is to restore all of the data from drive A: to drive C:. These settings can be changed.

Quick Format

The Quick Format program item enables you to format diskettes in a few seconds. The program uses the FORMAT with the /Q switch. The default setting is for drive A:, although you can change to another drive.

Format

Use the Format program item to format diskettes or fixed disks. The program uses the FORMAT command. The default setting is for drive A:, although you can change to another drive.

Undelete

The Undelete program item is used to recover deleted files. The MS-DOS UNDELETE command is called. The default setting uses the /LIST switch.

15.3.10 Protecting groups and programs

The MS-DOS Shell makes using MS-DOS a lot easier. One interesting feature of the MS-DOS Shell is an option to prevent users from accessing certain commands. This feature can be used to prevent data changes or destruction.

The following sections explain the different methods the MS-DOS Shell provides for protecting data.

Password

One easy method of protection is to install passwords for programs and program groups. When creating new program items or groups or making changes to existing programs or groups, enter a password in the Password text box of the Program Item Properties or Program Group Properties dialog box.

After a password is installed, the correct password must be entered before a group or program item can be activated from the MS-DOS Shell.

For example, all Utility programs can be protected as a group by assigning them to a group and then assigning the group a password.

A more practical example is to assign a password to the FORMAT command. This will prevent anyone who doesn't know the password from using FORMAT.

To remove the password, simply delete the expression from the appropriate line of the Properties dialog box. However, before you can make any changes, you must enter the password to gain access to the dialog box.

Unfortunately, a password offers only limited protection. All of the information about a group is stored in the DOSSHELL.INI file. The password is stored in an encrypted form. Also, someone could gain access to the protected file(s) by discovering the password.

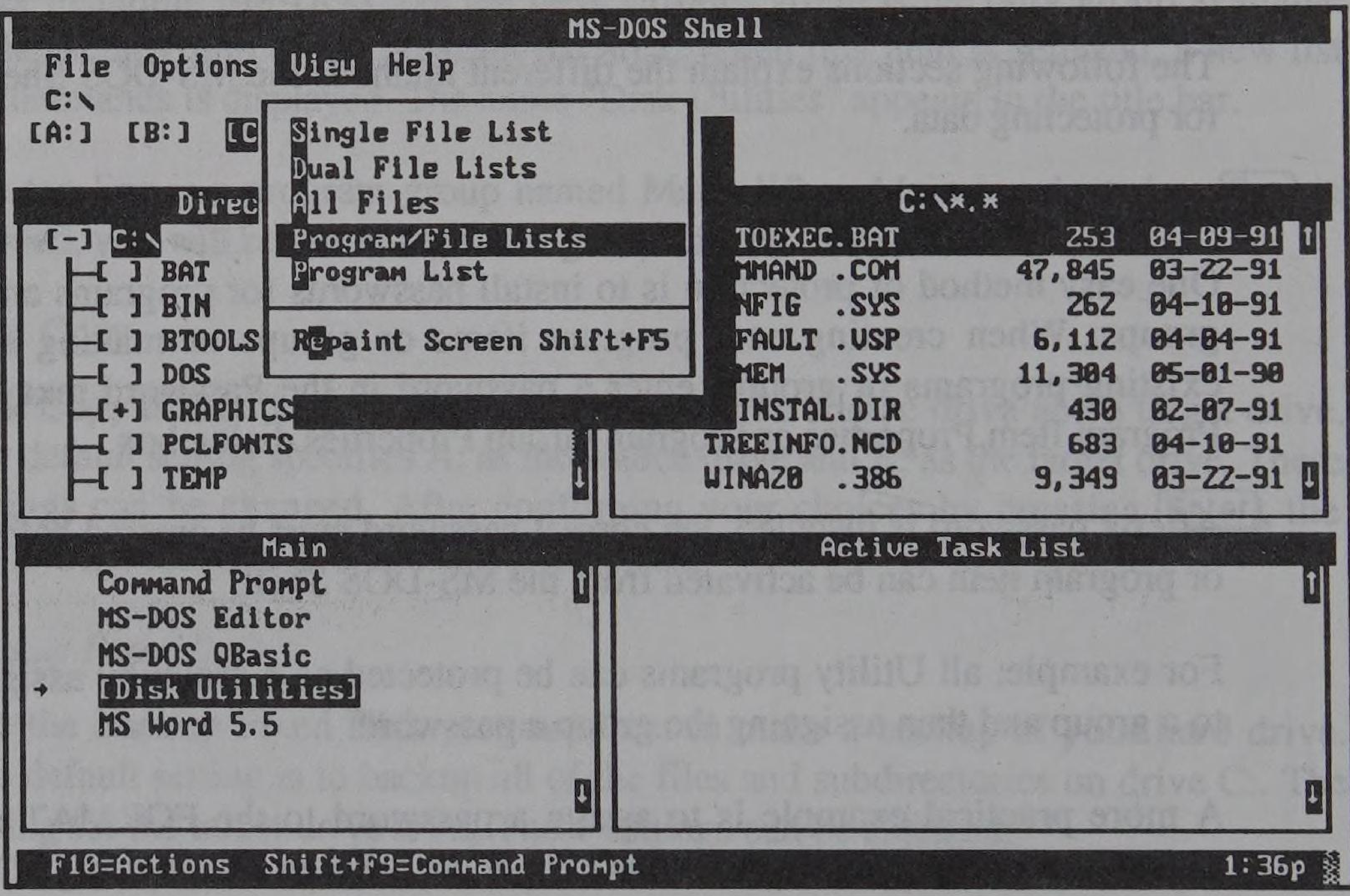
15.4 Working With Files And Directories

Before MS-DOS 5.0, the only easy methods available for working with files were supplied by third party developers. However, the Directory Tree/file list area of the MS-DOS Shell provides an easy solution to most of your file handling tasks.

The Directory Tree/file list area can be used to edit files, directories, diskettes and hard drives. These areas of the MS-DOS Shell can be used to create, rename, copy, compare and delete files. These are only some of the commands that are available.

15.4.1 The screen of the Directory Tree/file list area

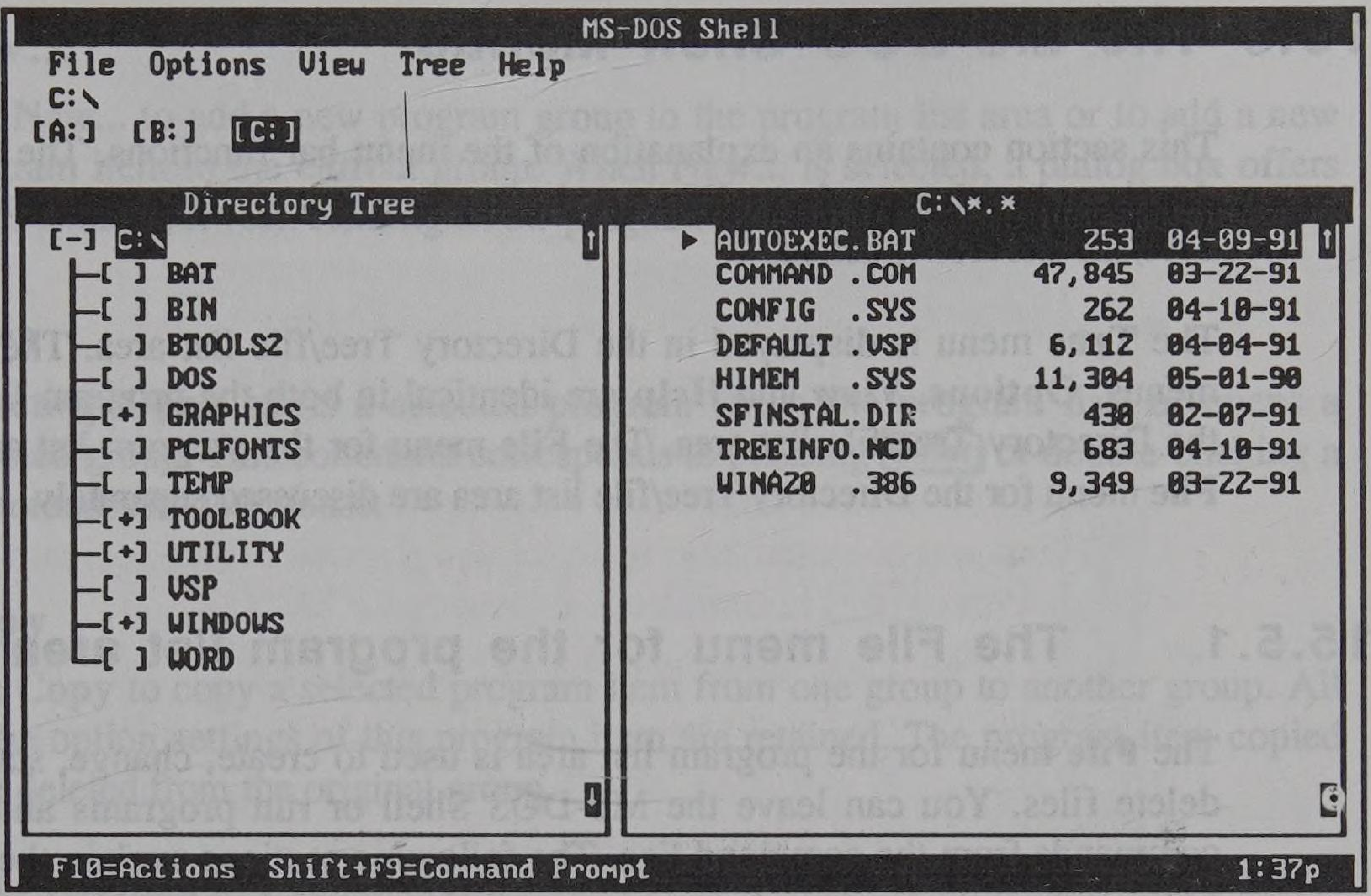
The **View** menu on the menu bar lets you display the Directory Tree/file list area as either one or two lists or display both the Directory Tree/file list area and the program list area on the screen simultaneously.


MS-DOS Shell
File Options View Help
C:\
[A:] [B:] [C
Single File List
Dual File Lists
All Files
Program/File Lists
Program List
Repaint Screen Shift+F5
Direc
C:\*.*
[-] C:\
[ ] BAT
[ ] BIN
[ ] BTOOLS
[ ] DOS
[+] GRAPHICS
[ ] PCLFONTS
[ ] TEMP
TOEXEC.BAT 253 04-09-91
MMAND .COM 47,845 03-22-91
NFIG .SYS 262 04-10-91
FAULT .USP 6,112 04-04-91
MEM .SYS 11,304 05-01-90
INSTAL.DIR 430 02-07-91
TREEINFO.NCD 683 04-10-91
WINA20 .386 9,349 03-22-91
Main
Command Prompt
MS-DOS Editor
MS-DOS QBasic
[Disk Utilities]
MS Word 5.5
Active Task List
F10=Actions Shift+F9=Command Prompt
1:36p

The View menu lets you customize the display modes of the MS-DOS Shell

If you choose **Single File List**, the Directory Tree/file list area fills the entire screen with one file list. The left side of the screen displays the Directory Tree while the right side of the screen displays the file list of the selected directory.

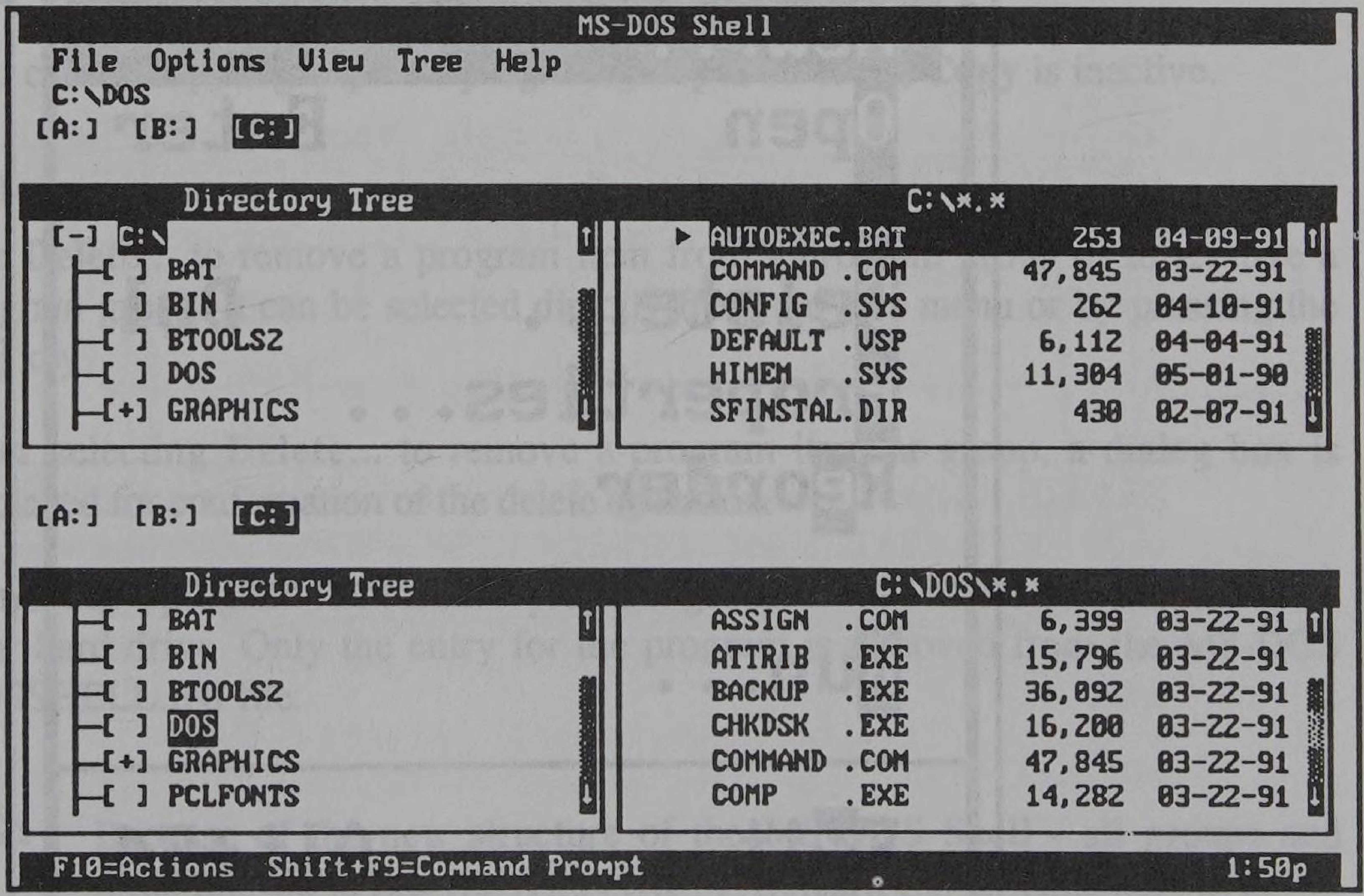
If **Dual File Lists** is selected, the Directory Tree/file list area is divided horizontally on the screen. This lets you display the contents and structure of two drives. The Directory Tree area is located on the left and the file list area of the current directory is on the right.



```
                              MS-DOS Shell
 File  Options  View  Tree  Help
 C:\
 [A:]  [B:]  [C:]

        Directory Tree                          C:\*.*
 [-] C:\                          ▶ AUTOEXEC.BAT        253  04-09-91
  ├[ ] BAT                          COMMAND .COM     47,845  03-22-91
  ├[ ] BIN                          CONFIG  .SYS        262  04-10-91
  ├[ ] BTOOLS2                      DEFAULT .VSP      6,112  04-04-91
  ├[ ] DOS                          HIMEM   .SYS     11,304  05-01-90
  ├[+] GRAPHICS                     SFINSTAL.DIR        430  02-07-91
  ├[ ] PCLFONTS                     TREEINFO.NCD        683  04-10-91
  ├[ ] TEMP                         WINA20  .386      9,349  03-22-91
  ├[+] TOOLBOOK
  ├[+] UTILITY
  ├[ ] VSP
  ├[+] WINDOWS
  └[ ] WORD

 F10=Actions  Shift+F9=Command Prompt                               1:37p
```

Single File List command from the View menu displays directories and files

The drive icons can be used to select which drive should be displayed. You can display two different drives or different directories on the same drive, as the following shows.



```
                              MS-DOS Shell
 File  Options  View  Tree  Help
 C:\DOS
 [A:]  [B:]  [C:]

        Directory Tree                          C:\*.*
 [-] C:\                          ▶ AUTOEXEC.BAT        253  04-09-91
  ├[ ] BAT                          COMMAND .COM     47,845  03-22-91
  ├[ ] BIN                          CONFIG  .SYS        262  04-10-91
  ├[ ] BTOOLS2                      DEFAULT .VSP      6,112  04-04-91
  ├[ ] DOS                          HIMEM   .SYS     11,304  05-01-90
  ├[+] GRAPHICS                     SFINSTAL.DIR        430  02-07-91

 [A:]  [B:]  [C:]

        Directory Tree                          C:\DOS\*.*
  ├[ ] BAT                          ASSIGN  .COM      6,399  03-22-91
  ├[ ] BIN                          ATTRIB  .EXE     15,796  03-22-91
  ├[ ] BTOOLS2                      BACKUP  .EXE     36,092  03-22-91
  ├[ ] DOS                          CHKDSK  .EXE     16,200  03-22-91
  ├[+] GRAPHICS                     COMMAND .COM     47,845  03-22-91
  ├[ ] PCLFONTS                     COMP    .EXE     14,282  03-22-91

 F10=Actions  Shift+F9=Command Prompt                               1:50p
```

Dual File Lists can display both the root directory and the DOS directory of C:

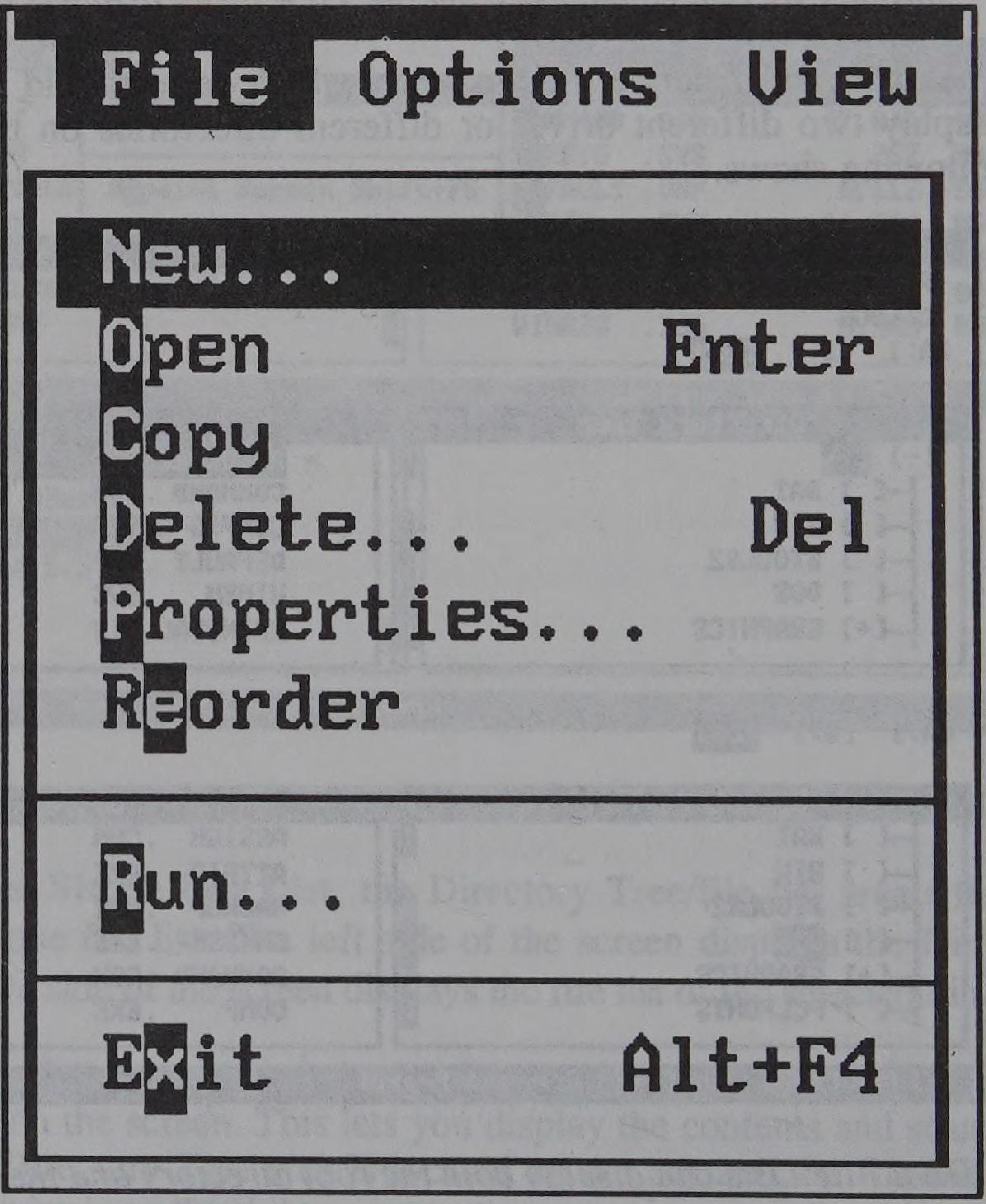
15.5 The MS-DOS Shell Menus

This section contains an explanation of the menu bar functions. The contents of the menu bar change, depending on whether the program list area or the Directory Tree/file list area is active.

The **Tree** menu is displayed in the Directory Tree/file list area. The remaining menus, **Options, View** and **Help** are identical in both the program list area and the Directory Tree/file list area. The **File** menu for the program list area and the **File** menu for the Directory Tree/file list area are discussed separately.

15.5.1 The File menu for the program list area

The **File** menu for the program list area is used to create, change, start, copy or delete files. You can leave the MS-DOS Shell or run programs and MS-DOS commands from the command line. The following sections explain the individual commands of the program list area.



The File menu for the program list area

New...

Use **New...** to add a new program group to the program list area or to add a new program item to the current group. When **New...** is selected, a dialog box offers you a choice between creating a new program item or a new program group.

Open

Selecting **Open** starts a selected program (from the program list) or opens a selected group. This command corresponds to pressing Enter or double-clicking a command with the mouse.

Copy

Use **Copy** to copy a selected program item from one group to another group. All of the option settings of this program item are retained. The program item copied isn't deleted from the original group.

After selecting this command, the following message is displayed on the status bar, at bottom of the screen: "Display the group to copy to, then press F2. Press Esc to cancel".

Use the cursor keys or the mouse to select the target group and open it with Enter. When the desired group appears on the screen, press F2 to copy the program item. Pressing Esc cancels the operation.

You cannot copy a group. If a program group is selected, **Copy** is inactive.

Delete...

Use **Delete...** to remove a program item from a program group or to remove a program group. It can be selected directly from the **File** menu or by pressing the Del key.

After selecting **Delete...** to remove a program item or group, a dialog box is displayed for confirmation of the delete operation.

Removing a program item from a program group doesn't delete the program from your hard drive. Only the entry for the program is removed from the MS-DOS DOSSHELL.INI file.

Because of the new structure of the MS-DOS Shell - all groups and programs are in one file (DOSSHELL.INI) - it's very dangerous to delete a group containing several program lines or subgroups. The deleted group is permanently removed, along with all of the entries. When the DOSSHELL.INI file is changed, a backup copy isn't made.

Properties...

Use **Properties...** to view existing program lines or group lines. The options are the same as for New. For more information about this item, refer to the sections on installing and changing programs and setting up groups.

Reorder

Use **Reorder** to change the position of a selected program group or program item within the active group. After this command is selected, the following message appears in the bottom screen line: "Select location to move to, then press Enter". Use the mouse or cursor keys to select and position the line containing the title to change. Press Enter to complete the process.

Run...

When **Run...** is activated, a dialog box appears in which you can enter an MS-DOS command or program name. Then press Enter to execute the command or program. After the command or program is completed, a message informs you to press any key to return to the MS-DOS Shell.

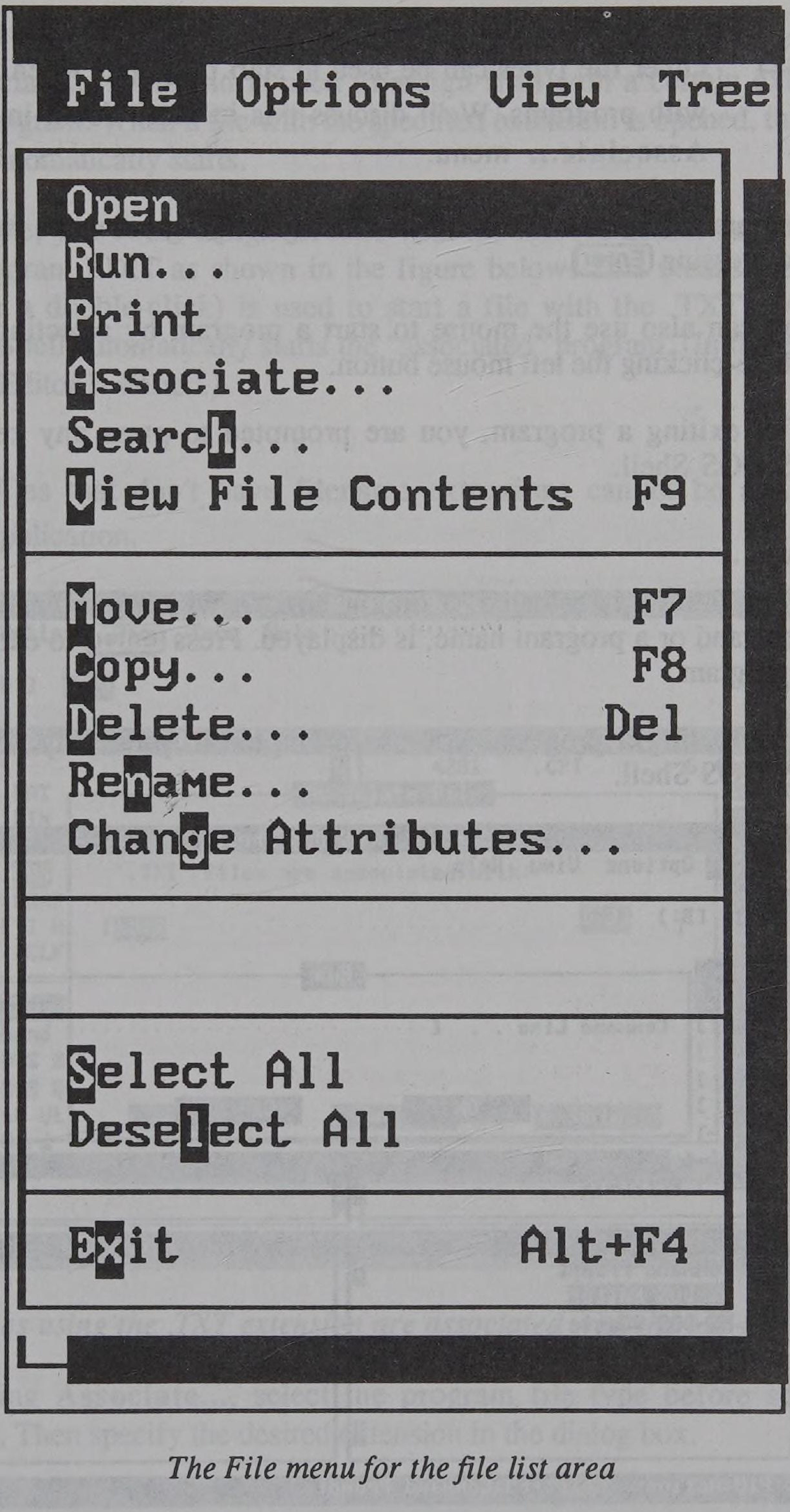
Don't run any TSR programs from the MS-DOS Shell. These types of programs could cause the MS-DOS Shell to crash.

Exit

The last command of the menu is **Exit**. When selected, the MS-DOS Shell terminates without any additional prompt. The MS-DOS Shell can also be ended by typing Alt + F4 or by pressing F3.

15.5.2 The File menu for the file list area

The **File** menu of the Directory Tree/file list area can be used to create, delete, copy and rename files and directories. It can also start programs, view file contents, search for files, assign programs to file groups and more. The following sections explain the different commands of the **File** menu of the Directory Tree/file list area.


File
Options
View
Tree
Open
Run...
Print
Associate...
Search...
View File Contents F9
Move... F7
Copy... F8
Delete... Del
Rename...
Change Attributes...
Select All
Deselect All
Exit Alt+F4

The File menu for the file list area

Open

Use **Open** to start programs. All programs or executable files have .COM, .EXE and .BAT filename extensions. Attempting to open any other types of files displays an error message.

Other file types can be used to start programs when they are associated with programs. We'll discuss this in more detail in the section on the **Associate...** menu.

Programs can also be started by selecting the program name from the file list area and pressing Enter.

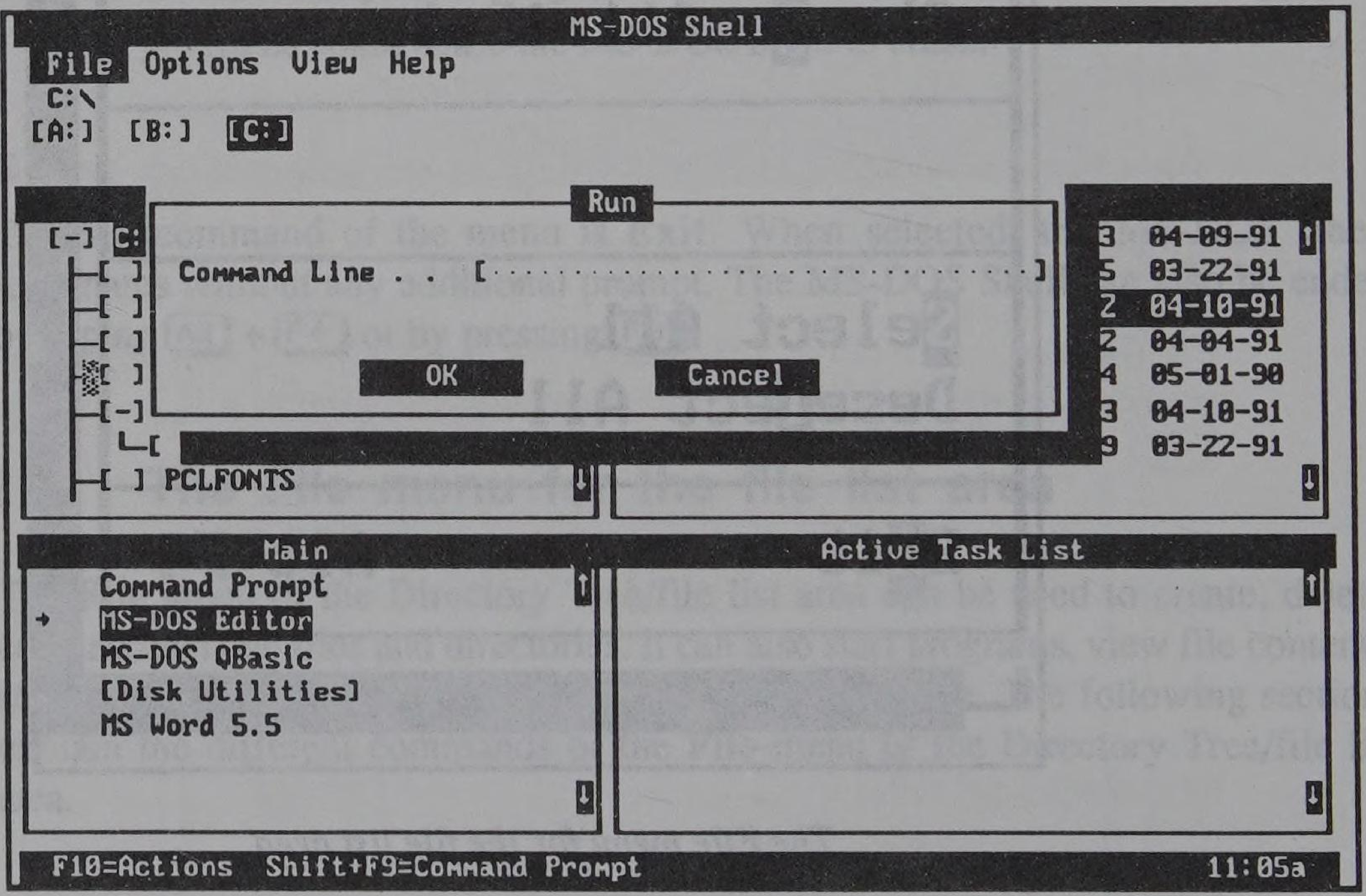
You can also use the mouse to start a program by selecting the filename and double-clicking the left mouse button.

After exiting a program, you are prompted to press any key to return to the MS-DOS Shell.

Run...

When **Run...** is selected, a dialog box, in which you can enter an MS-DOS command or a program name, is displayed. Press Enter to execute the command or program.

After exiting a program, you are prompted to press any key to return to the MS-DOS Shell.


MS-DOS Shell
File Options View Help
C:\
[A:] [B:] [C:]
Run
Command Line . . [..............................]
OK
Cancel
[-] C:
[ ] PCLFONTS
04-09-91
03-22-91
04-10-91
04-04-91
05-01-90
04-10-91
03-22-91
Main
Command Prompt
MS-DOS Editor
MS-DOS QBasic
[Disk Utilities]
MS Word 5.5
Active Task List
F10=Actions Shift+F9=Command Prompt
11:05a

A dialog box for entering the command is displayed when Run... is activated

Print

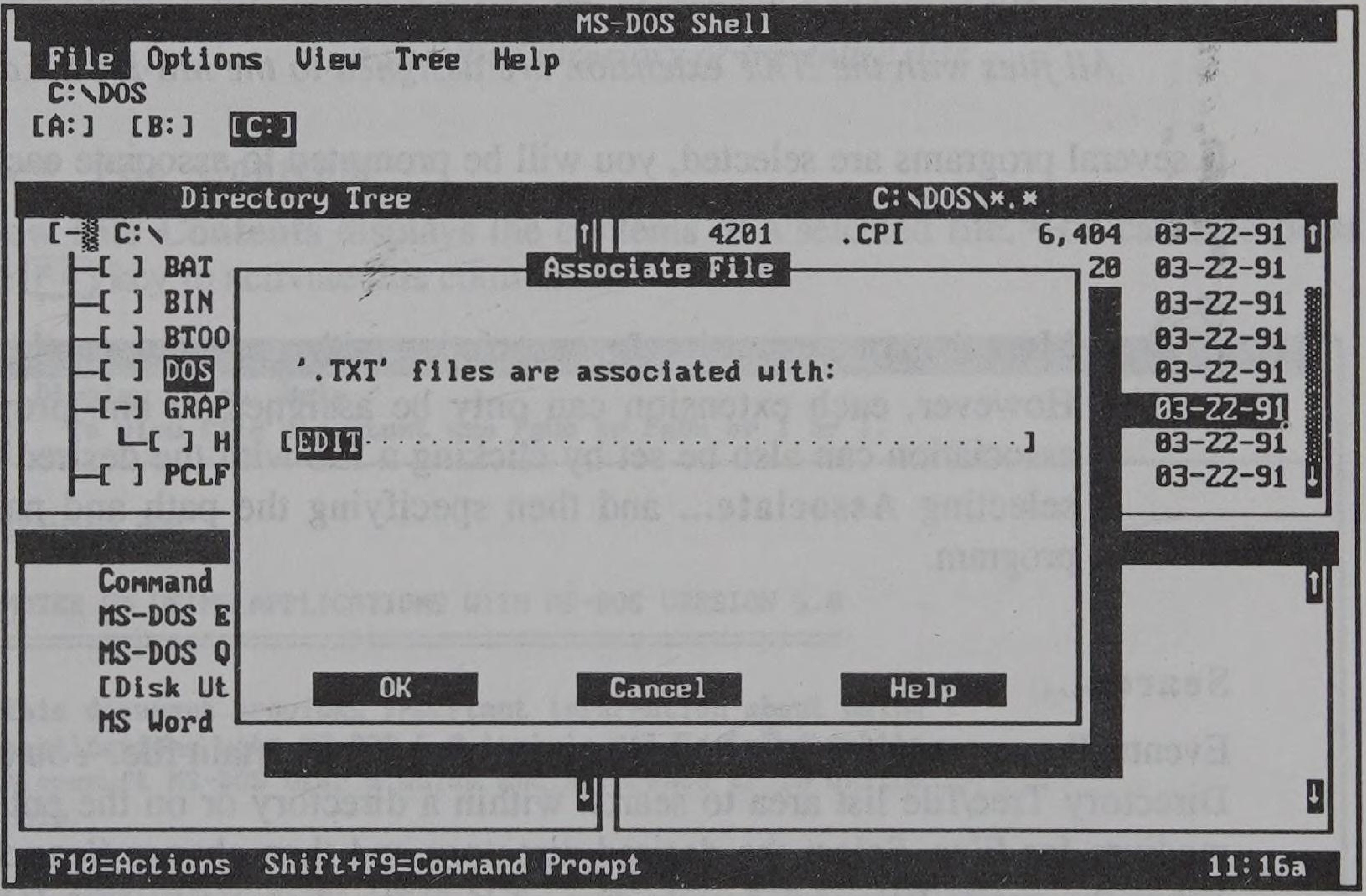
Use **Print** to print the contents of selected files to an attached printer. The printing operation takes place in the background.

Associate...

The **Associate...** command is used to assign files with a certain extension to a specific program. When a file with the specified extension is opened, the associated program automatically starts.

For example, you could assign all files with the .TXT extension to the MS-DOS Editor program EDIT as shown in the figure below. This means that whenever OPEN (or a double-click) is used to start a file with the .TXT extension, the MS-DOS Shell automatically starts the "associated" program. (In the example, the MS-DOS Editor is started.)

Files that don't have filename extensions cannot be assigned to an application.

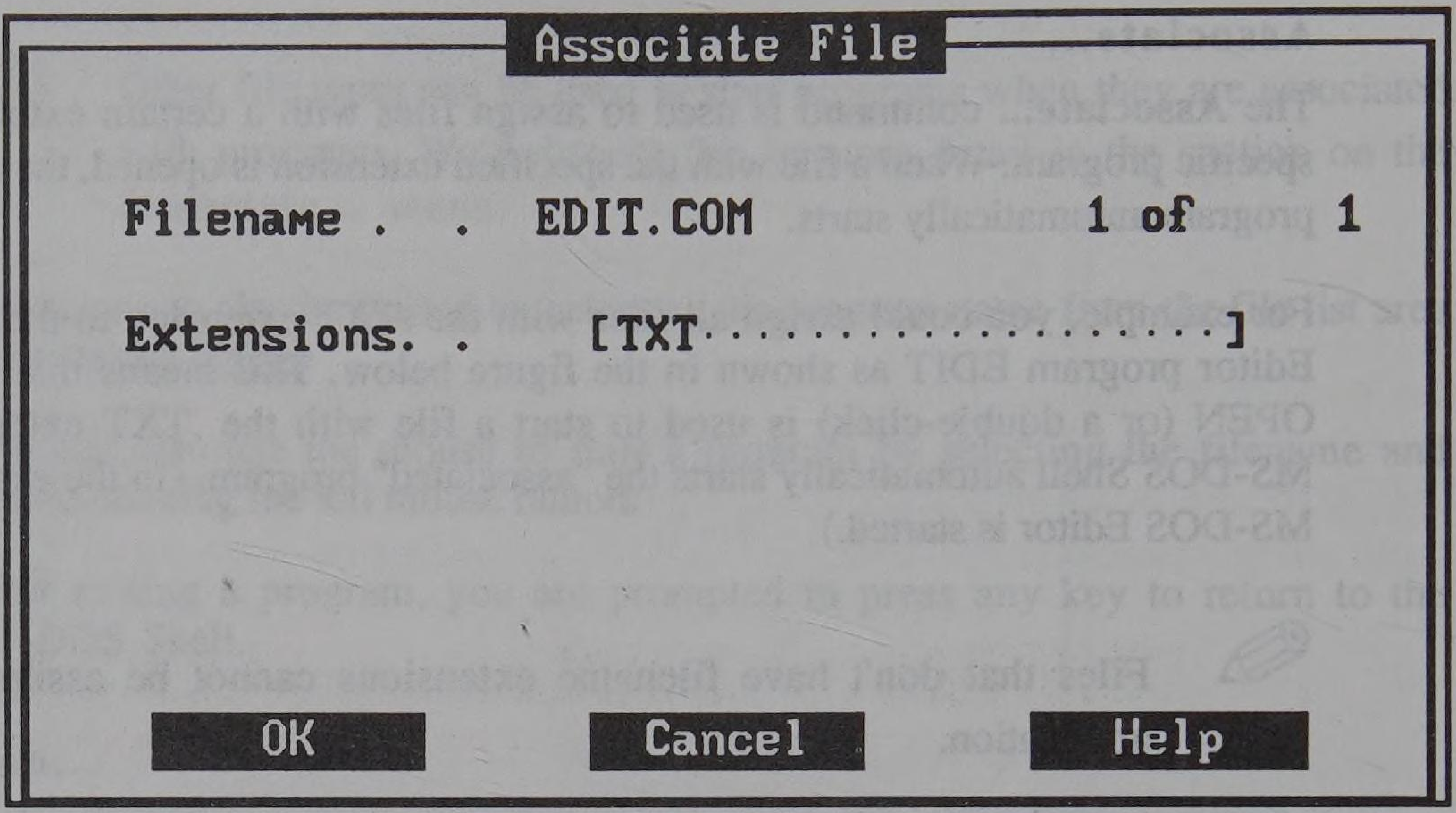

MS-DOS Shell
File Options View Tree Help
C:\DOS
[A:] [B:] [C:]
Directory Tree
C:\DOS\*.*
[-] C:\
[ ] BAT
[ ] BIN
[ ] BTOO
[ ] DOS
[-] GRAP
[ ] H
[ ] PCLF
4201 .CPI 6,404 03-22-91
20 03-22-91
03-22-91
03-22-91
03-22-91
03-22-91
03-22-91
03-22-91
Command
MS-DOS E
MS-DOS Q
[Disk Ut
MS Word
Associate File
OK
Cancel
Help
F10=Actions Shift+F9=Command Prompt
11:16a

'.TXT' files are associated with:

[EDIT..]

All files using the .TXT extension are associated with the MS-DOS Editor

When using **Associate...**, select the program file type before selecting this command. Then specify the desired extension in the dialog box.



```
Filename . .   EDIT.COM              1 of   1

Extensions. .  [TXT····················]
```

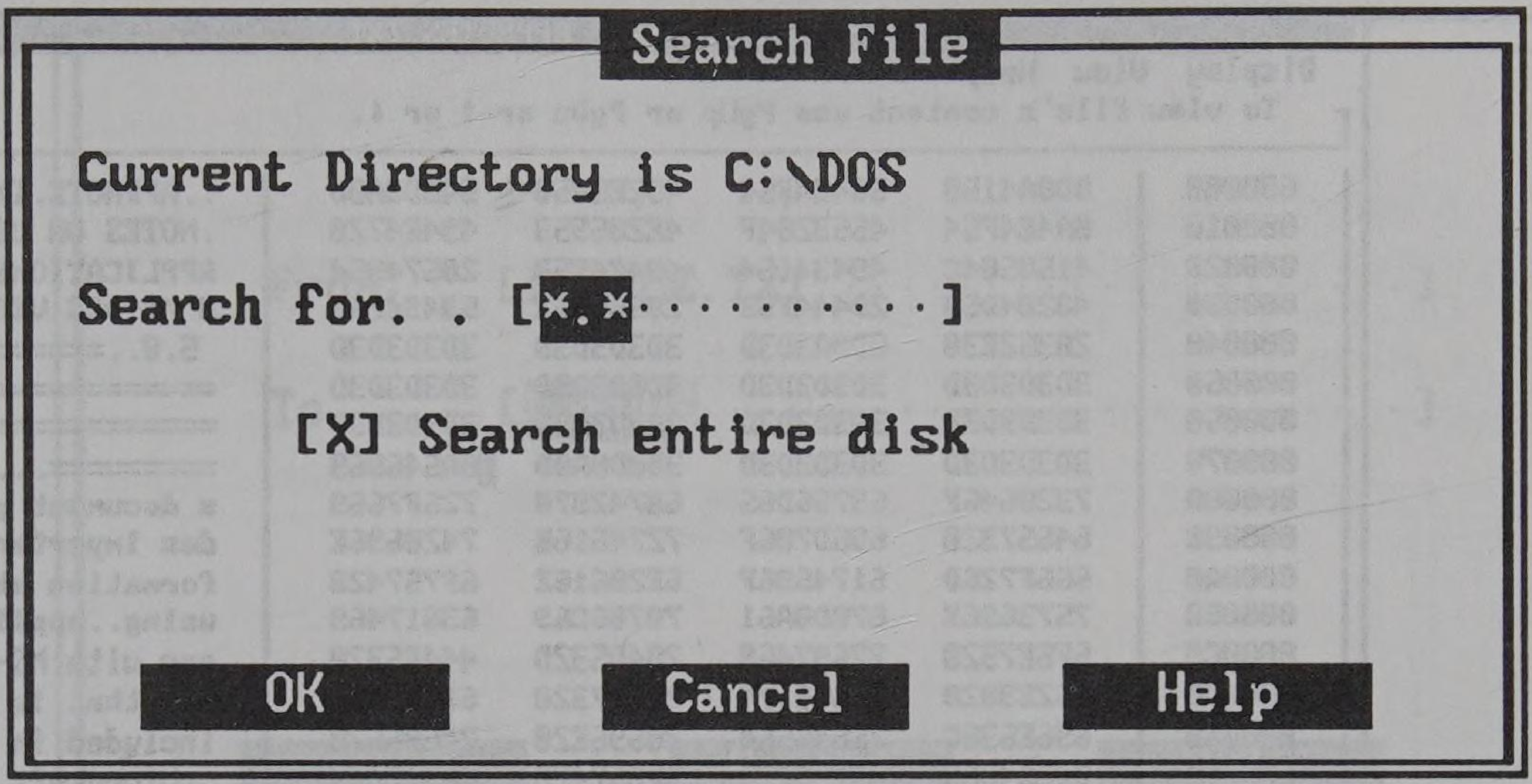
All files with the .TXT extension are assigned to the MS-DOS Editor

If several programs are selected, you will be prompted to associate each program with file types.

More than one extension for association with a program can be specified. However, each extension can only be assigned to one program. The association can also be set by clicking a file with the desired extension, selecting **Associate...** and then specifying the path and name of the program.

Search...

Eventually you may forget which directory contains a certain file. You can ask the Directory Tree/file list area to search within a directory or on the entire storage medium for files. Select the desired directory and then choose **Search....** Enter either the entire filename or just part of it in the dialog box that is displayed. Wildcards (* and ?) are accepted.

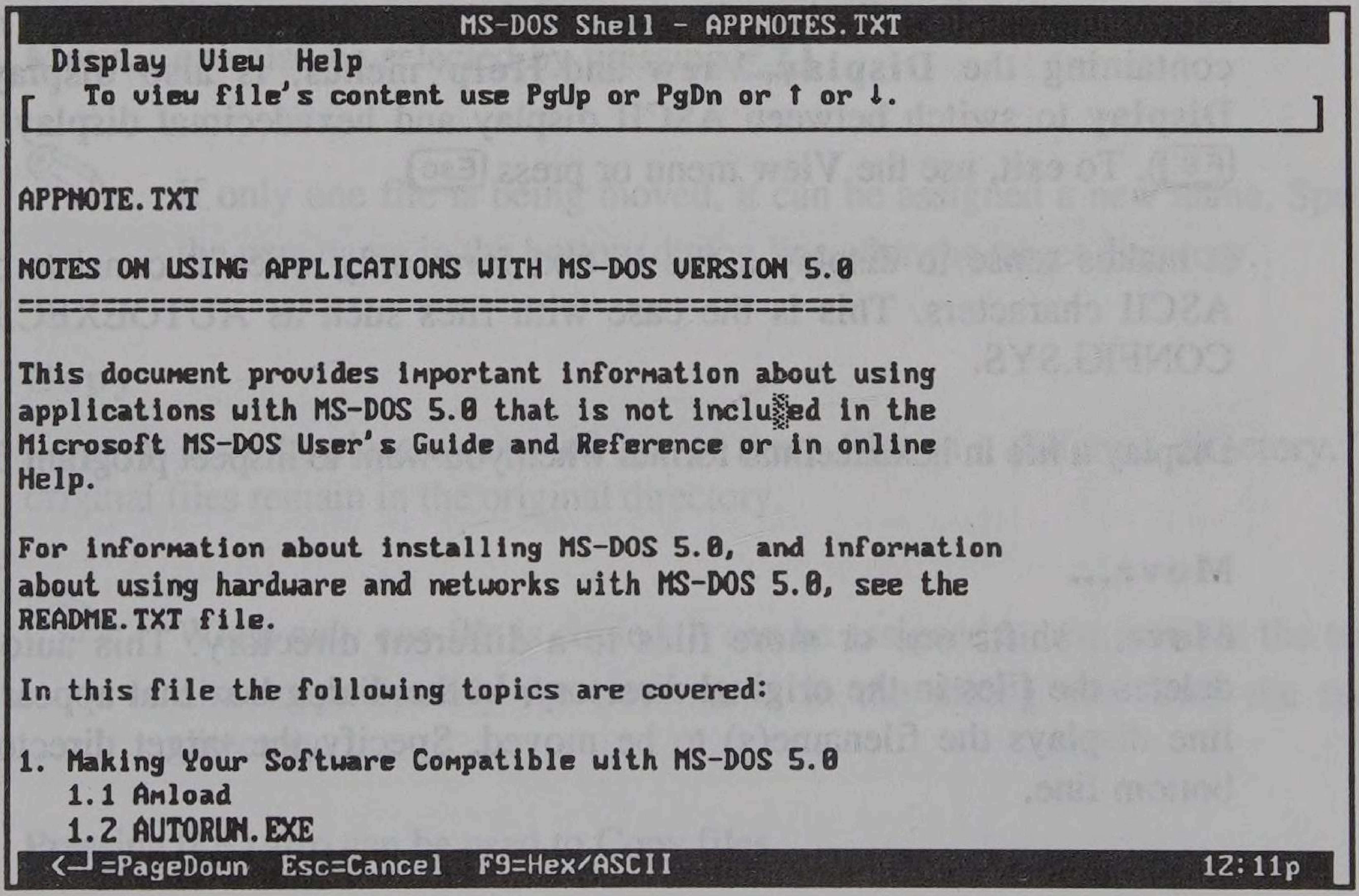


```
Search File

Current Directory is C:\DOS

Search for. . [*.*··········]

[X] Search entire disk
```

The Search File dialog box lets you search for a specified file or files in either a specified directory or the entire disk

View File Contents

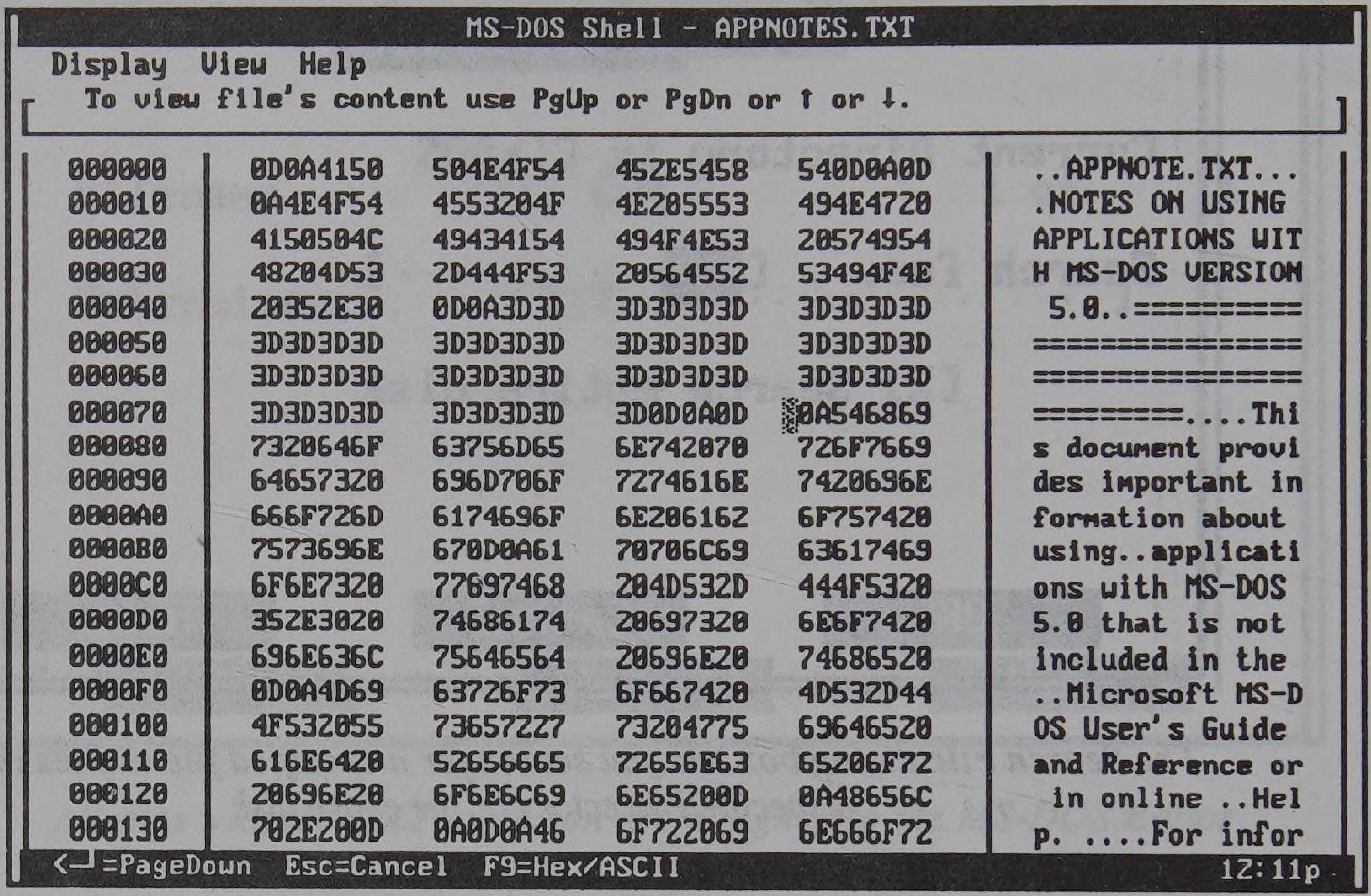
View File Contents displays the contents of a selected file. You can also press the F9 key to activate this command.



```
MS-DOS Shell - APPNOTES.TXT
Display  View  Help
  To view file's content use PgUp or PgDn or ↑ or ↓.

APPNOTE.TXT

NOTES ON USING APPLICATIONS WITH MS-DOS VERSION 5.0
===================================================

This document provides important information about using
applications with MS-DOS 5.0 that is not included in the
Microsoft MS-DOS User's Guide and Reference or in online
Help.

For information about installing MS-DOS 5.0, and information
about using hardware and networks with MS-DOS 5.0, see the
README.TXT file.

In this file the following topics are covered:

1. Making Your Software Compatible with MS-DOS 5.0
   1.1 Amload
   1.2 AUTORUN.EXE
 <-┘=PageDown  Esc=Cancel  F9=Hex/ASCII                    12:11p
```

Using View File Contents to display a file as ASCII text

When the file is displayed, pressing F9 switches between ASCII and hexadecimal display modes. Use PgUp and PgDn to scroll through the pages of the file or use ↑ and ↓ to scroll line by line through the file. Press Esc to return to file management.



```
 To view file's content use PgUp or PgDn or ↑ or ↓.

000000   0D0A4150   504E4F54   452E5458   540D0A0D   ..APPNOTE.TXT...
000010   0A4E4F54   4553204F   4E205553   494E4720   .NOTES ON USING
000020   4150504C   49434154   494F4E53   20574954   APPLICATIONS WIT
000030   48204D53   2D444F53   20564552   53494F4E   H MS-DOS VERSION
000040   20352E30   0D0A3D3D   3D3D3D3D   3D3D3D3D    5.0..==========
000050   3D3D3D3D   3D3D3D3D   3D3D3D3D   3D3D3D3D   ================
000060   3D3D3D3D   3D3D3D3D   3D3D3D3D   3D3D3D3D   ================
000070   3D3D3D3D   3D3D3D3D   3D0D0A0D   0A546869   =========....Thi
000080   7320646F   63756D65   6E742070   726F7669   s document provi
000090   64657320   696D706F   7274616E   7420696E   des important in
0000A0   666F726D   6174696F   6E206162   6F757420   formation about
0000B0   7573696E   670D0A61   70706C69   63617469   using..applicati
0000C0   6F6E7320   77697468   204D532D   444F5320   ons with MS-DOS
0000D0   352E3020   74686174   20697320   6E6F7420   5.0 that is not
0000E0   696E636C   75646564   20696E20   74686520   included in the
0000F0   0D0A4D69   63726F73   6F667420   4D532D44   ..Microsoft MS-D
000100   4F532055   73657227   73204775   69646520   OS User's Guide
000110   616E6420   52656665   72656E63   65206F72   and Reference or
000120   20696E20   6F6E6C69   6E65200D   0A48656C    in online ..Hel
000130   702E200D   0A0D0A46   6F722069   6E666F72   p. ....For infor
```

Pressing F9 while using View File Contents toggles between ASCII and hexadecimal display modes

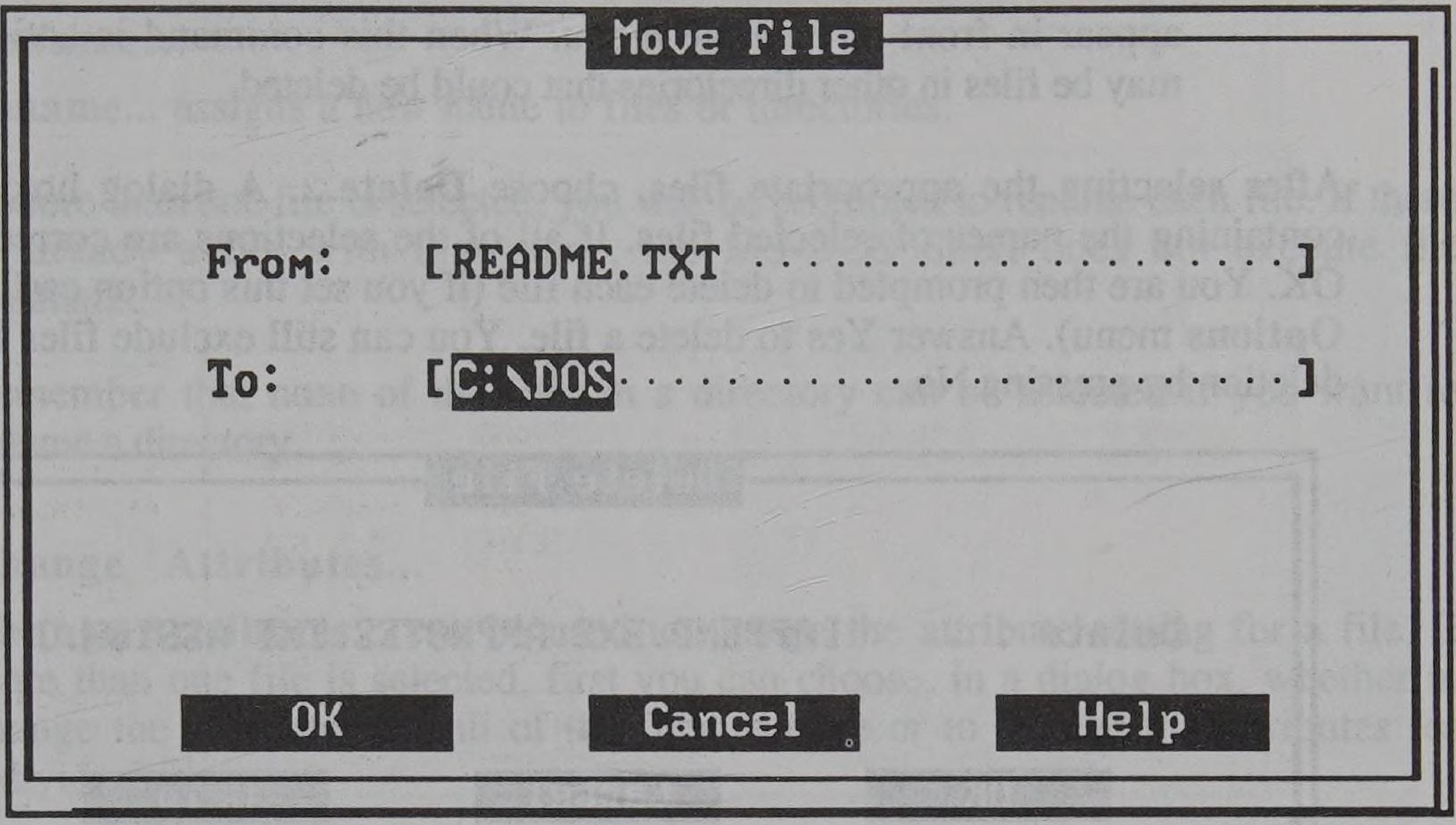
The top screen line displays the name of the file being viewed. A new menu bar, containing the **Display**, **View** and **Help** menus, is also displayed. Use **Display** to switch between ASCII display and hexadecimal display (or press F9). To exit, use the **View** menu or press Esc.

It makes sense to display a file in text form only when it consists of legible ASCII characters. This is the case with files such as AUTOEXEC.BAT and CONFIG.SYS.

Display a file in hexadecimal format when you want to inspect program files.

Move...

Move... shifts one or more files to a different directory. This automatically deletes the files in the original directory. In the dialog box that appears, the top line displays the filename(s) to be moved. Specify the target directory in the bottom line.



```
From:    [README.TXT ·····················]

To:      [C:\DOS··························]
```

Enter the source file and the target directory or disk in the Move File dialog box

Since **Move...** deletes the original file, be very careful when using it.

Move... can also be selected by pressing F7.

If only one file is being moved, it can be assigned a new name. Specify the new name in the bottom dialog line after the target directory.

Copy

Copy is used to place a copy of one or more files in a different directory. The original files remain in the original directory.

When only one file is copied, it can be assigned a new name in the target directory. Specify the new name in the dialog box with the target directory.

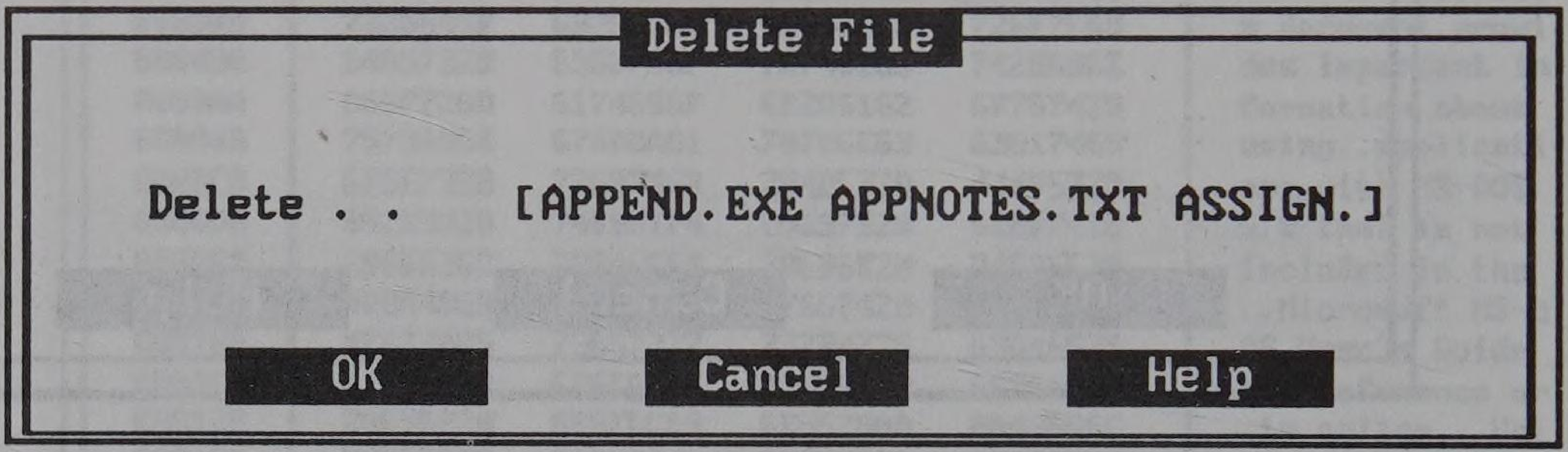
Pressing F8 also can be used to Copy files.

Delete...

Delete... erases all selected files.

It's possible that the **Select Across Directories** command in the **Options** menu is activated. If this command is selected, a black dot will appear in front of it in the menu. When this command is active, there may be files in other directories that could be deleted.

After selecting the appropriate files, choose **Delete....** A dialog box appears containing the names of selected files. If all of the selections are correct, select OK. You are then prompted to delete each file (if you set this option earlier in the **Options** menu). Answer Yes to delete a file. You can still exclude files from the deletion by pressing No.

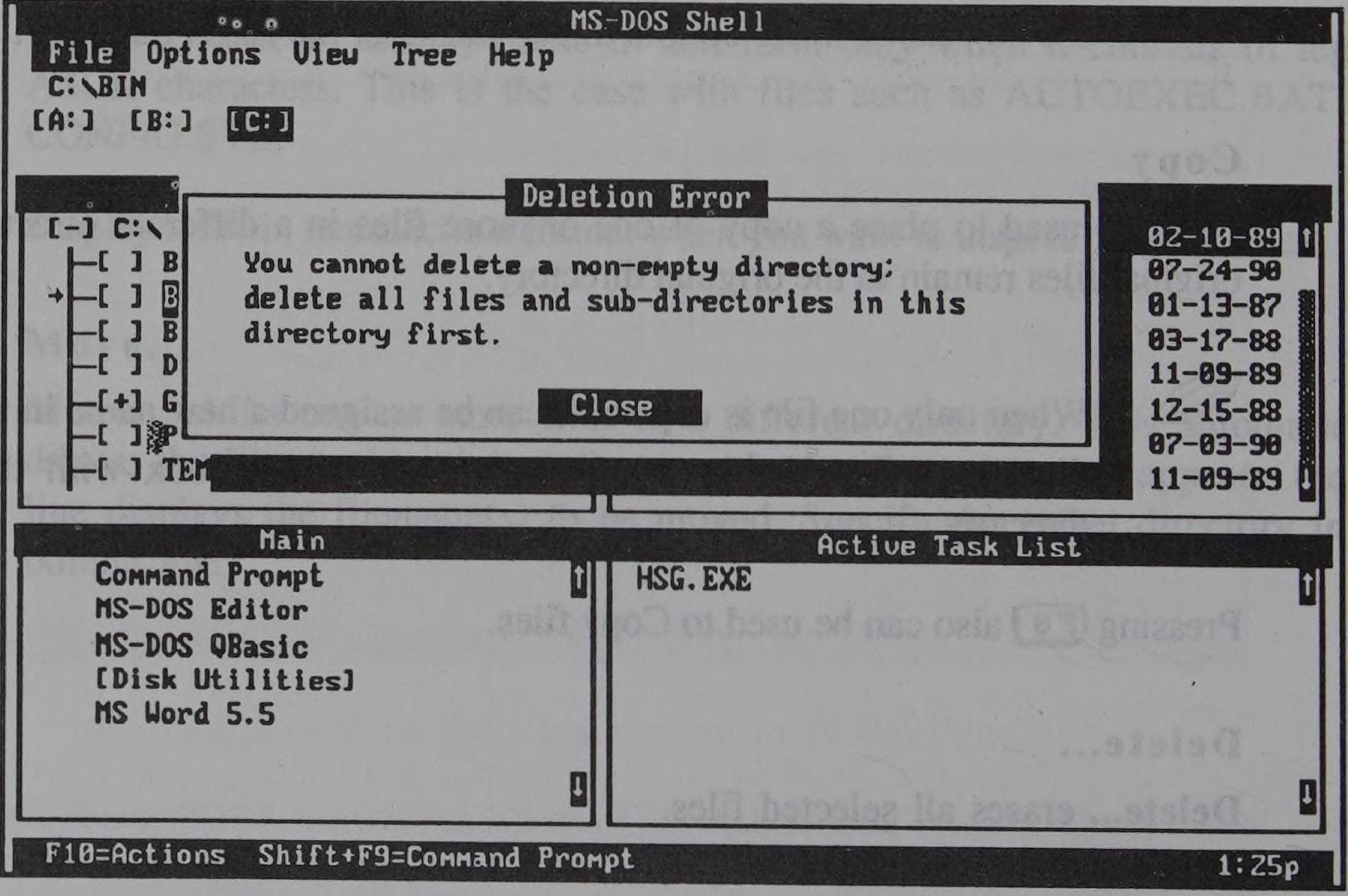

Delete File
OK
Cancel
Help

Delete . .   [APPEND.EXE APPNOTES.TXT ASSIGN.]

APPEND.EXE, APPNOTES.TXT and ASSIGN.COM selected for deletion

If you have previously switched off Confirm on Delete from the **Confirmation...** command of the **Options** menu, the confirmation dialog box won't be displayed. The selected files will be immediately deleted.

If you selected a directory instead of a file, the (empty) directory can be deleted after you answer a prompt to confirm your intent. If the directory is not empty, a message, informing you of the error, is displayed.


MS-DOS Shell
File
Options
View
Tree
Help
C:\BIN
[A:]
[B:]
[C:]
[-] C:\
[ ] B
[ ] B
[ ] B
[ ] D
[+] G
[ ] P
[ ] TEM
02-10-89
07-24-90
01-13-87
03-17-88
11-09-89
12-15-88
07-03-90
11-09-89
Deletion Error
Close
Main
Command Prompt
MS-DOS Editor
MS-DOS QBasic
[Disk Utilities]
MS Word 5.5
Active Task List
HSG.EXE
F10=Actions
Shift+F9=Command Prompt
1:25p

You cannot delete a non-empty directory; delete all files and sub-directories in this directory first.

Deleting a directory which contains files generates this error message

Rename...

Rename... assigns a new name to files or directories.

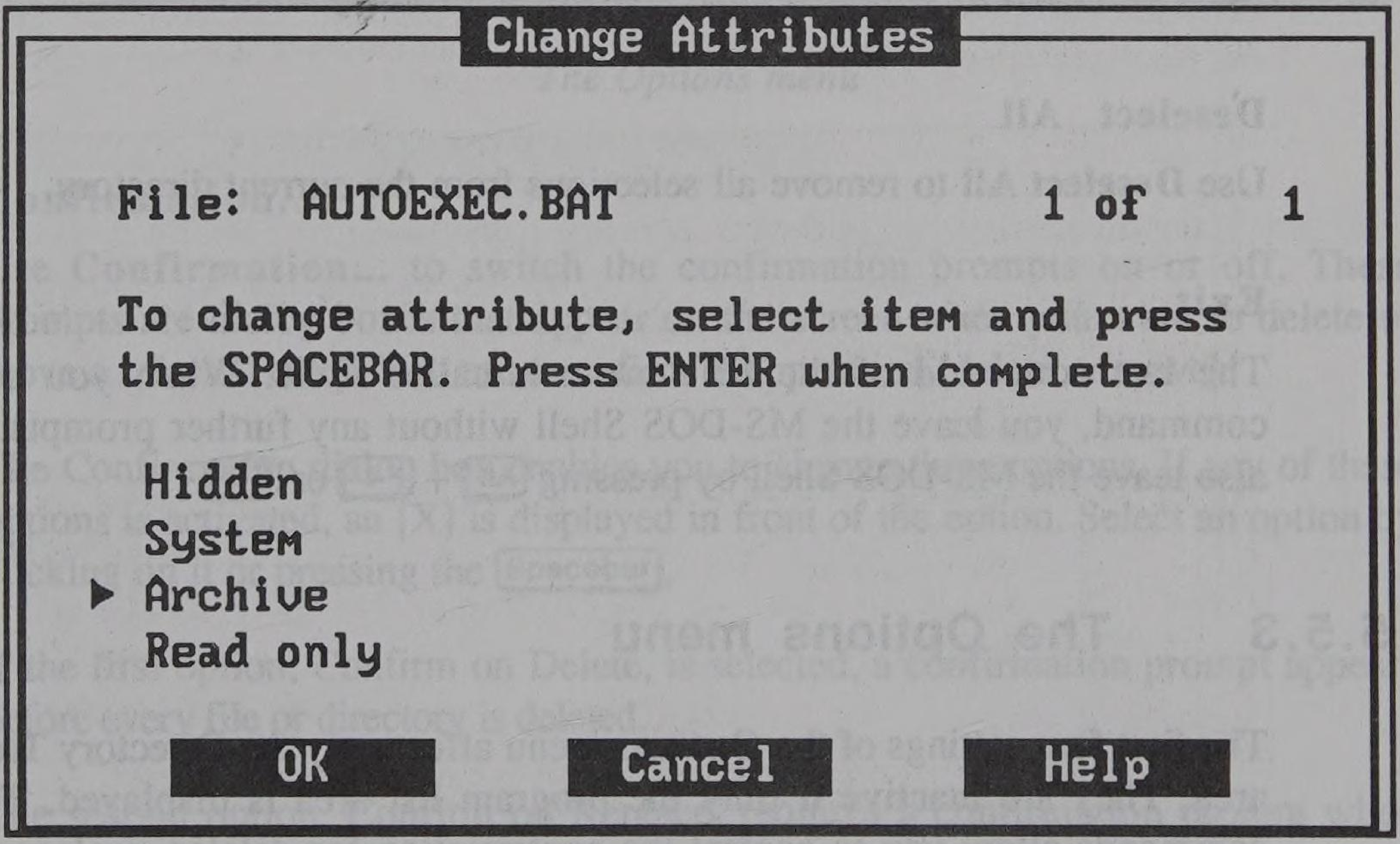
If more than one file is selected, you will be prompted to rename each file. If there is already a file with that name, the MS-DOS Shell does not execute the command.

Remember that none of the files in a directory can be selected if you want to rename a directory.

Change Attributes...

Change Attributes... can be used to change the attribute setting for a file. If more than one file is selected, first you can choose, in a dialog box, whether to change the attributes for all of the files at once or to change the attributes for individual files.

You can specify more than one attribute by selecting the attributes to change with the ↑ and ↓ cursor keys and pressing the Spacebar to select them or by using the mouse. A mark, which appears to the left of each attribute, indicates whether or not a file attribute is set.



Change Attributes

File: AUTOEXEC.BAT 1 of 1

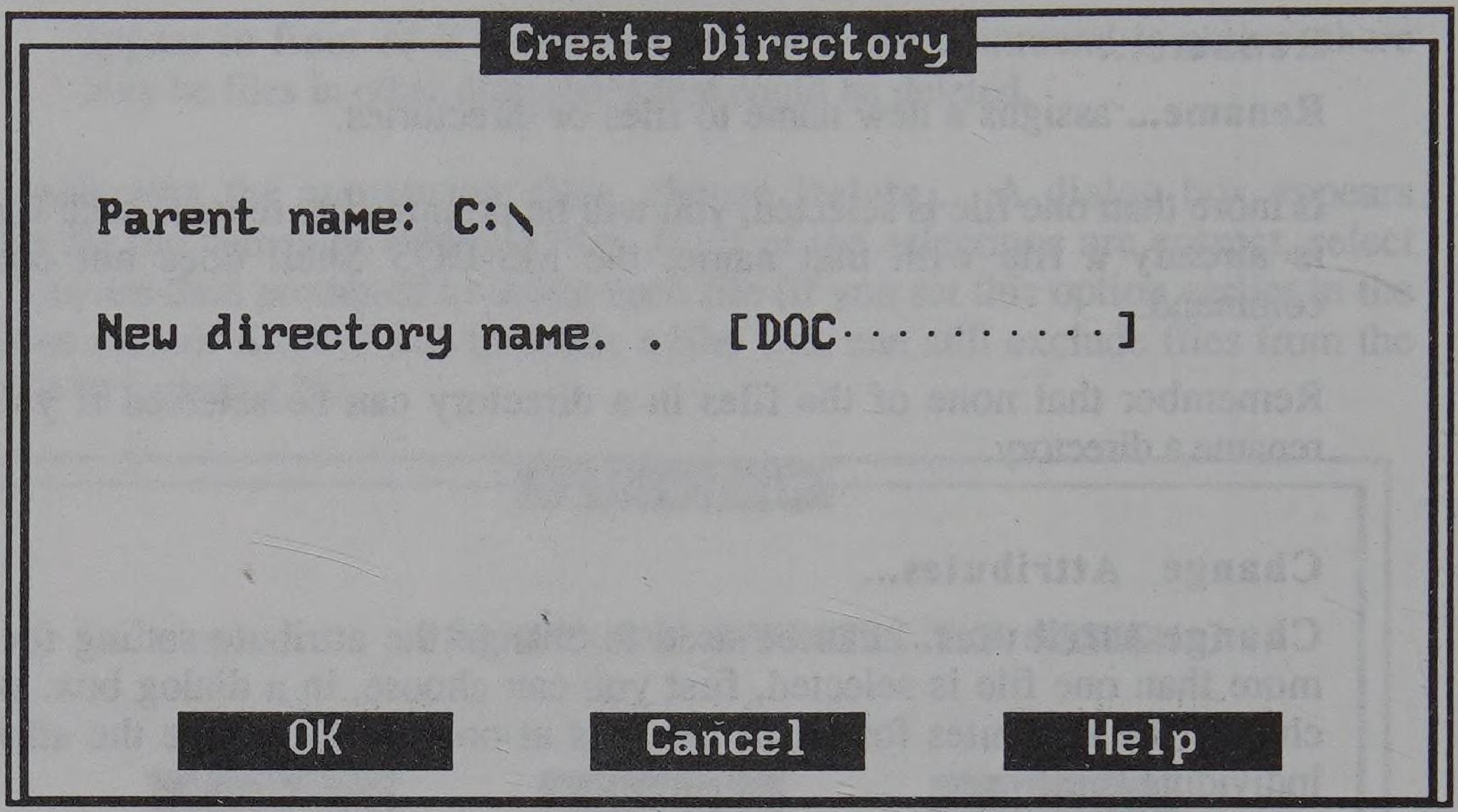
To change attribute, select item and press the SPACEBAR. Press ENTER when complete.

Hidden
System
▶ Archive
Read only

The Archive Bit is already set for AUTOEXEC.BAT

Create Directory...

To use the **Create Directory...** command to create a new directory, the Directory Tree area must be active. The new directory is always a subdirectory of the current (selected) directory. The current directory is displayed in the top line of the Create directory dialog box. Specify the name of the new directory in the next line. Use the same method to assign filenames.



```
Parent name: C:\

New directory name. .  [DOC··········]
```

Preparing to create a new directory named DOC using Create Directory

Select All

Use **Select All** to select all of the files of a directory.

Deselect All

Use **Deselect All** to remove all selections from the current directory.

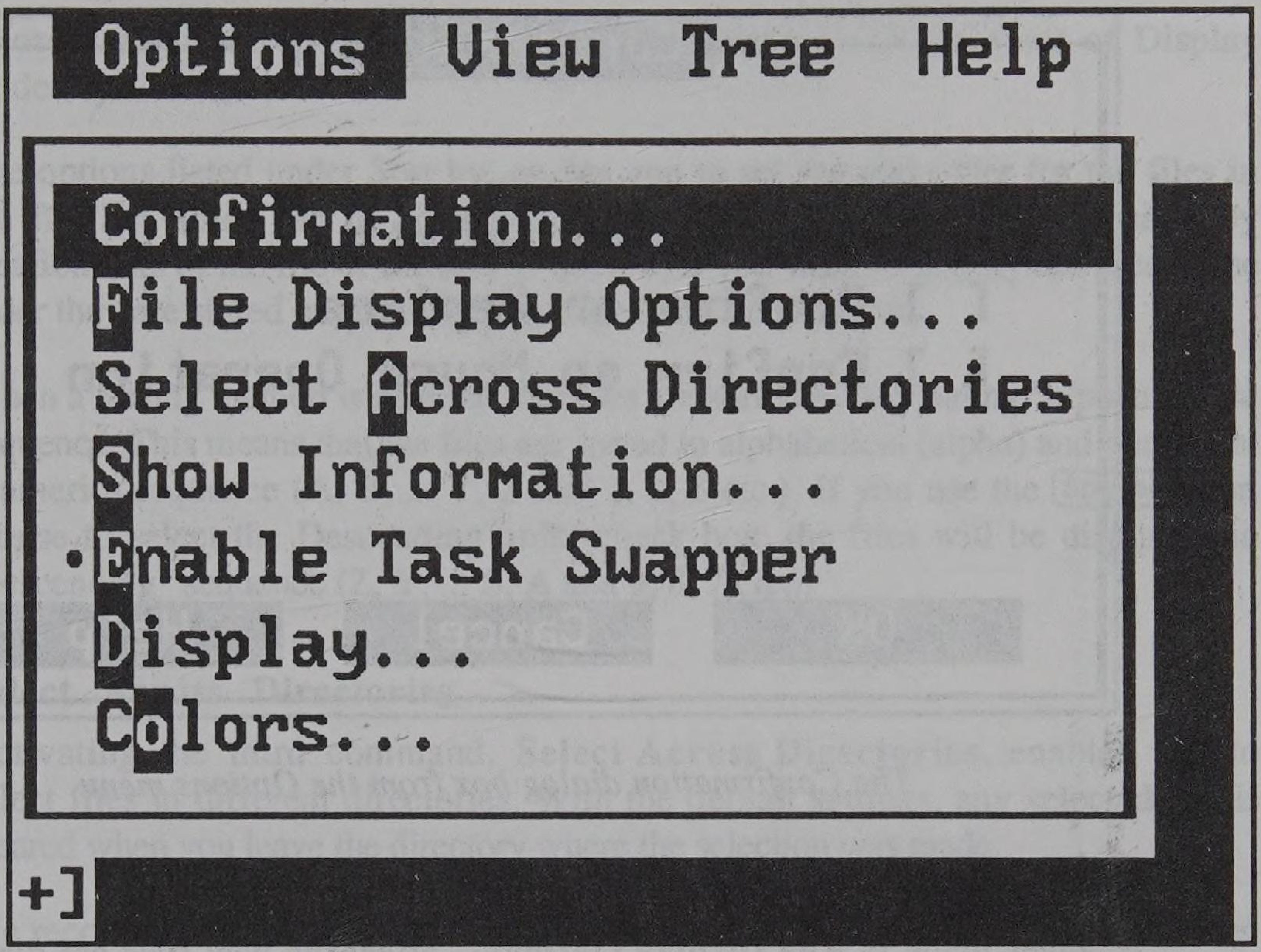
Exit

The last command of the **File** menu is called **Exit**. When you select this command, you leave the MS-DOS Shell without any further prompts. You can also leave the MS-DOS Shell by pressing Alt + F4 or F3.

15.5.3 The Options menu

The first four settings of the **Options** menu affect only the Directory Tree/file list area. They are inactive if only the program list area is displayed. These four commands allow you to control the confirmation for deletes, replacements and mouse operations. They also enable you to set the file display options, switch the selection option for more than one directory on and off and display detailed information on files, directories and drives.

These settings can switch the Task Swapper on and off, select the video display and make different settings for color and monochrome monitors.


Options
View
Tree
Help
Confirmation...
File Display Options...
Select Across Directories
Show Information...
•Enable Task Swapper
Display...
Colors...
+]

The Options menu

Confirmation...

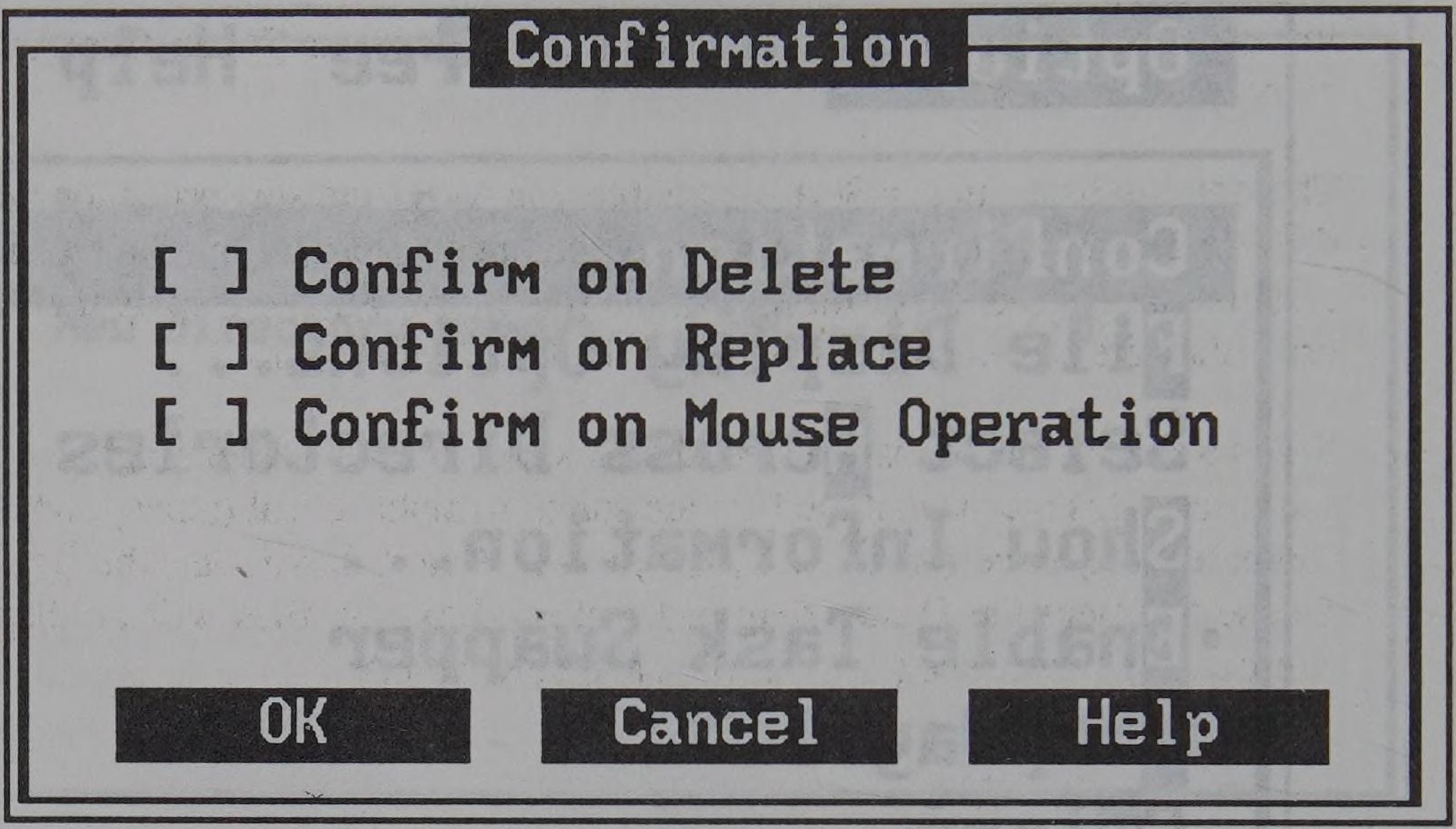
Use **Confirmation...** to switch the confirmation prompts on or off. These prompts are dialog boxes that appear on the screen when you want to delete or move a selected entry. You must confirm the operation (delete or move).

The Confirmation dialog box enables you to change three options. If any of these options is activated, an [X] is displayed in front of the option. Select an option by clicking on it or pressing the Spacebar.

If the first option, Confirm on Delete, is selected, a confirmation prompt appears before every file or directory is deleted.

The second option, Confirm on Replace, requires a confirmation prompt when copying and moving files.

Confirm on Mouse Operation, the third option, displays a confirmation prompt before copying or moving files while holding down the mouse button.

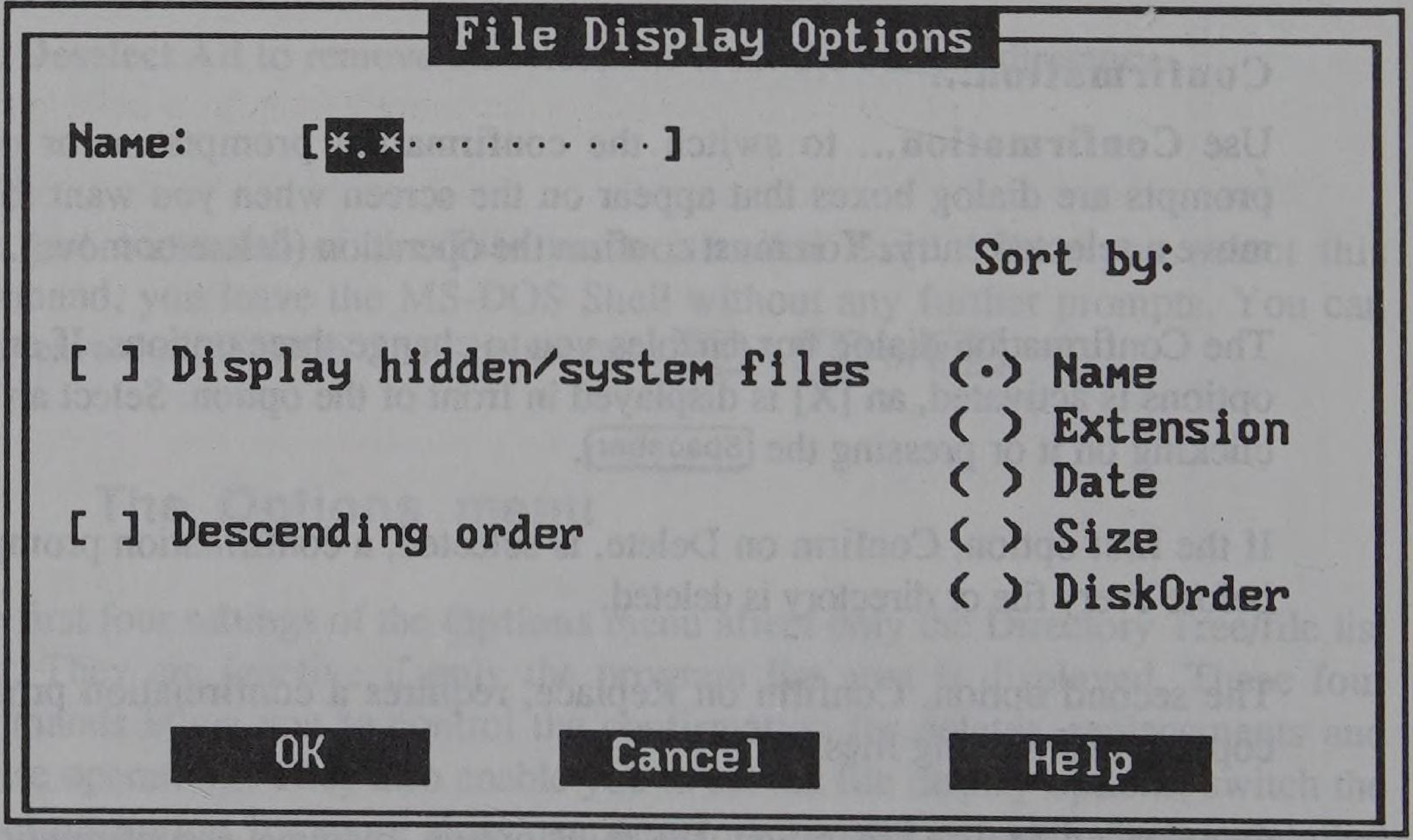


[ ] Confirm on Delete
[ ] Confirm on Replace
[ ] Confirm on Mouse Operation

The Confirmation dialog box from the Options menu

File Display Options...

Changes made in **File Display Options...** affect the way files are displayed in the file list of the Directory Tree/file list area.



Name: [*.*]

Sort by:

[ ] Display hidden/system files

[ ] Descending order

(•) Name
( ) Extension
( ) Date
( ) Size
( ) DiskOrder

Setting the file display options

In the Name: text box you can determine which files in the file list will be displayed. The default setting is *.*. This setting means that all files in the current directory are displayed. Other specifications can be made to limit the files displayed. Wildcard entries such as "*" and "?" are accepted.

In order to display hidden and system files, place an [X] in front of Display hidden/system files.

The options listed under Sort by: enable you to set the sort order for the files in the file list. Files can be sorted in alphanumeric order, by file extension, by creation date of the file or the size of the file. If you want to display the files in the order they are stored on the disk, select the DiskOrder option.

When a sorting method is selected, the files are sorted in "ascending" alphanumeric sequence. This means that the files are sorted in alphabetical (alpha) and numerical (numeric) sequence (A, B, ... Y, Z and 1, 2, 3 etc.). If you use the Spacebar or mouse to select the Descending order check box, the files will be displayed in "descending" sequence (Z, Y, ... B, A and 9, 8, 7 etc.).

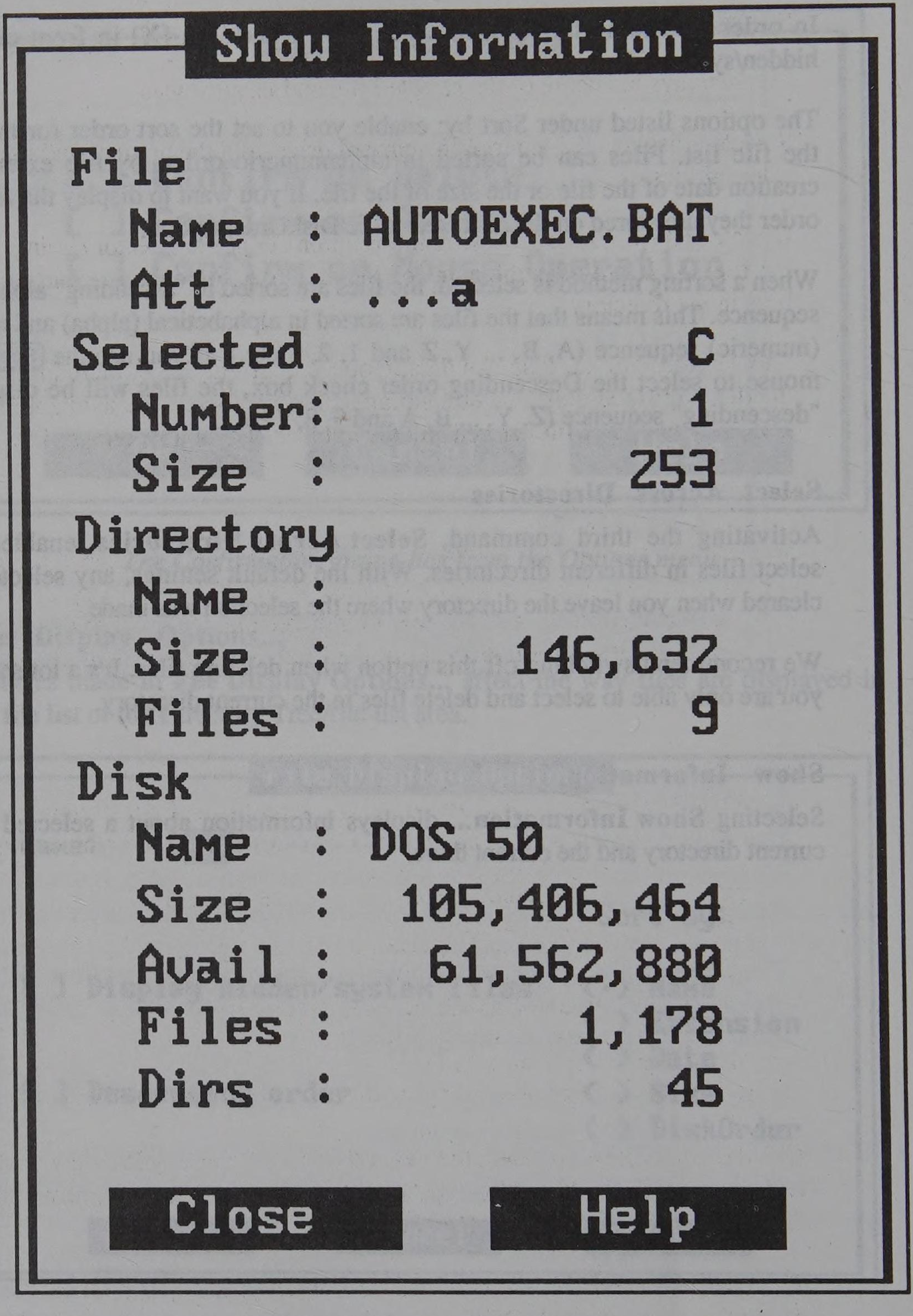
Select Across Directories

Activating the third command, **Select Across Directories**, enables you to select files in different directories. With the default settings, any selected file is cleared when you leave the directory where the selection was made.

We recommend switching off this option when deleting files. It's a lot safer when you are only able to select and delete files in the current directory.

Show Information...

Selecting **Show Information...** displays information about a selected file, the current directory and the current drive.



```
File
  Name   : AUTOEXEC.BAT
  Att    : ...a
Selected            C
  Number:           1
  Size   :        253
Directory
  Name   :
  Size   :    146,632
  Files  :          9
Disk
  Name   : DOS 50
  Size   : 105,406,464
  Avail  :  61,562,880
  Files  :       1,178
  Dirs   :          45
```

Show Information dialog box with information about the selected file (AUTOEXEC.BAT), the current directory (the root directory) and the current drive (C:)

Show Information... displays the following information:

File displays the name of the selected file and its attributes.

Selected displays the letter of the current drive, and the number and total size of all selected files.

Directory displays the following information about the directory being displayed in the file list. It provides the name of the directory, its total size and the number of files in the directory.

Show Information... displays data about the current drive, including the name of the diskette/fixed disk, its total storage capacity (Size), the available free storage area, the number of files stored and the number of available directories.

Enable Task Swapper

Selecting **Enable Task Swapper** switches the Task Swapper on and off. To activate this command, click once or select it with the cursor key and press Enter. A mark is placed in front of the text to indicate that the Task Swapper is switched on. More detailed information about the Task Swapper is available elsewhere in this book.

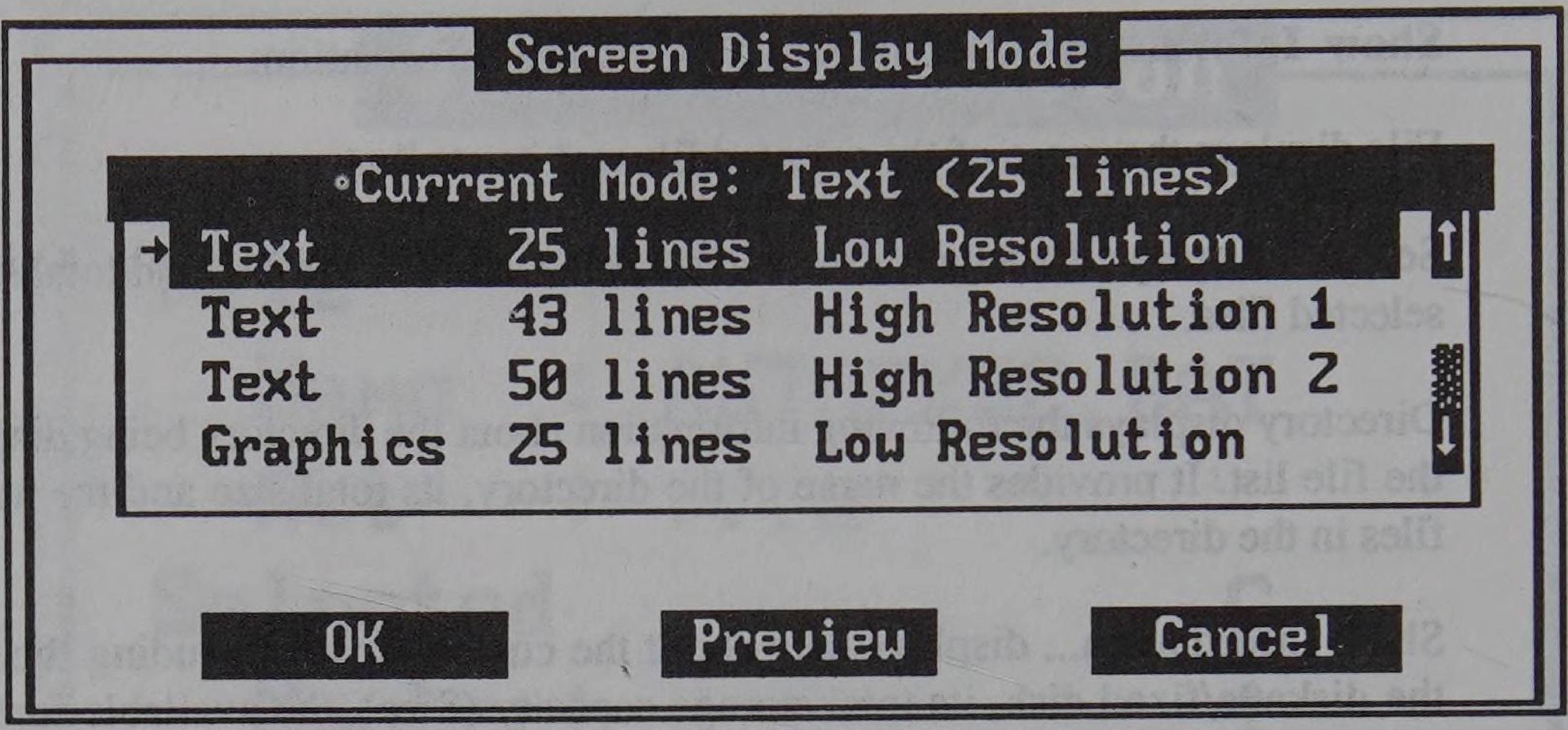
Display...

Use **Display...** to choose one of eight screen displays if you have a VGA graphics card. Three options are in text mode and five are in graphics mode:

```
Text        Low Resolution          (25 Lines)
Text        High Resolution 1       (43 Lines)
Text        High Resolution 2       (50 Lines)
Graphics    Low Resolution          (25 Lines)
Graphics    Medium Resolution 1     (30 Lines)
Graphics    Medium Resolution 2     (34 Lines)
Graphics    High Resolution 1       (43 Lines)
Graphics    High Resolution 2       (60 Lines)
```

Use the cursor keys to select the desired option. Selecting Preview enables you to test the screen display. After you've selected a screen mode, press Enter. This changes the screen display to the one selected.

The Hercules monochrome graphics card limits your options. There are two kinds of text resolution with 25 lines each: graphics resolution with 25 lines and graphics resolution with 43 lines.



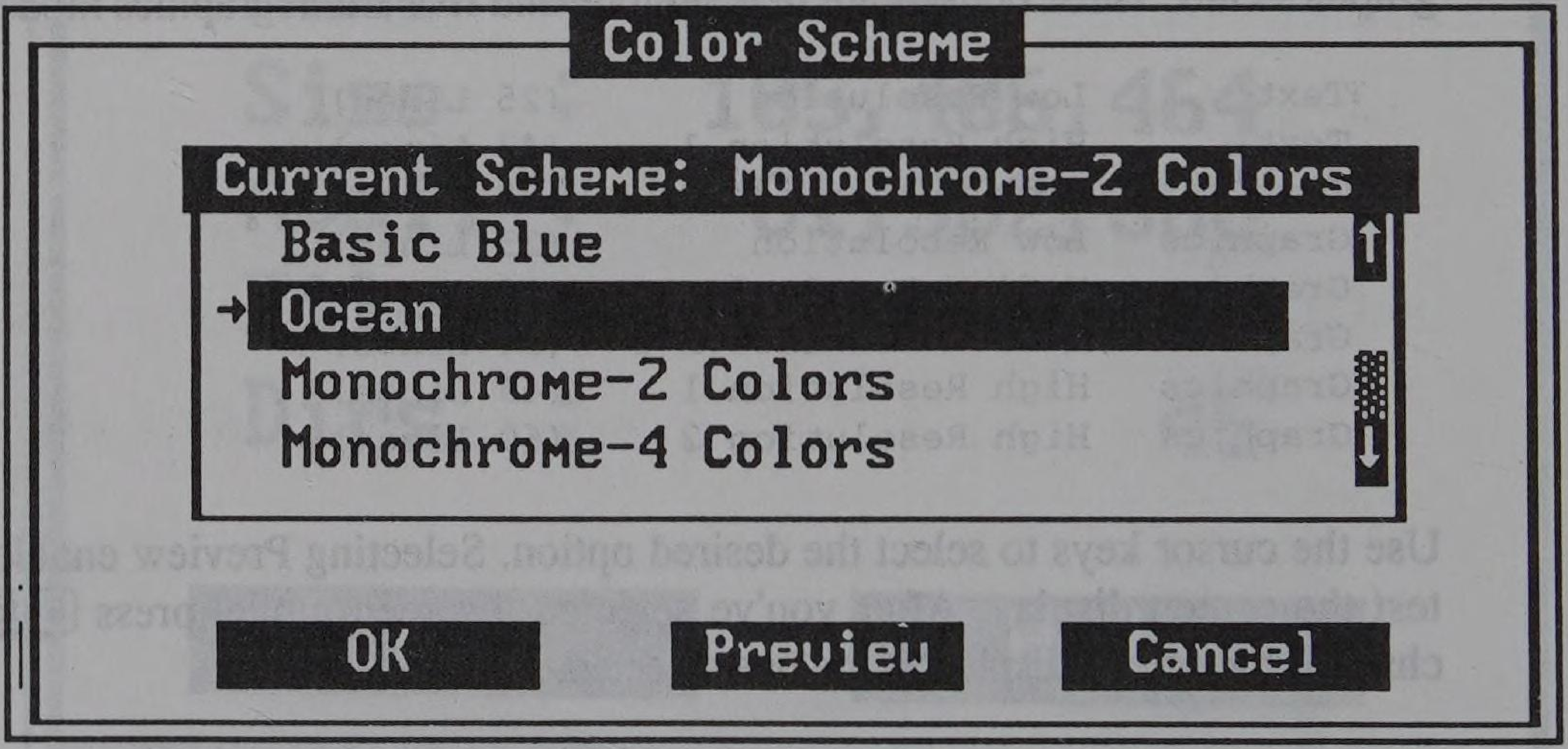
```
•Current Mode: Text (25 lines)
→ Text        25 lines  Low Resolution
  Text        43 lines  High Resolution 1
  Text        50 lines  High Resolution 2
  Graphics    25 lines  Low Resolution
```

Selecting the screen display mode

Colors...

If you have a color monitor, the **Colors...** command lets you choose from four preset settings. Two of the settings are monochrome and two are color. The Color Scheme dialog box displays the active color scheme in the top dialog line.

Use the ↑ cursor key to move to the settings. Then use ↑ and ↓ to select a setting and press Enter. This activates the selected color display. If you like the display, accept it by pressing Enter. If you don't like the display, press Esc to activate the previous display.



```
Current Scheme: Monochrome-2 Colors
  Basic Blue
→ Ocean
  Monochrome-2 Colors
  Monochrome-4 Colors
```

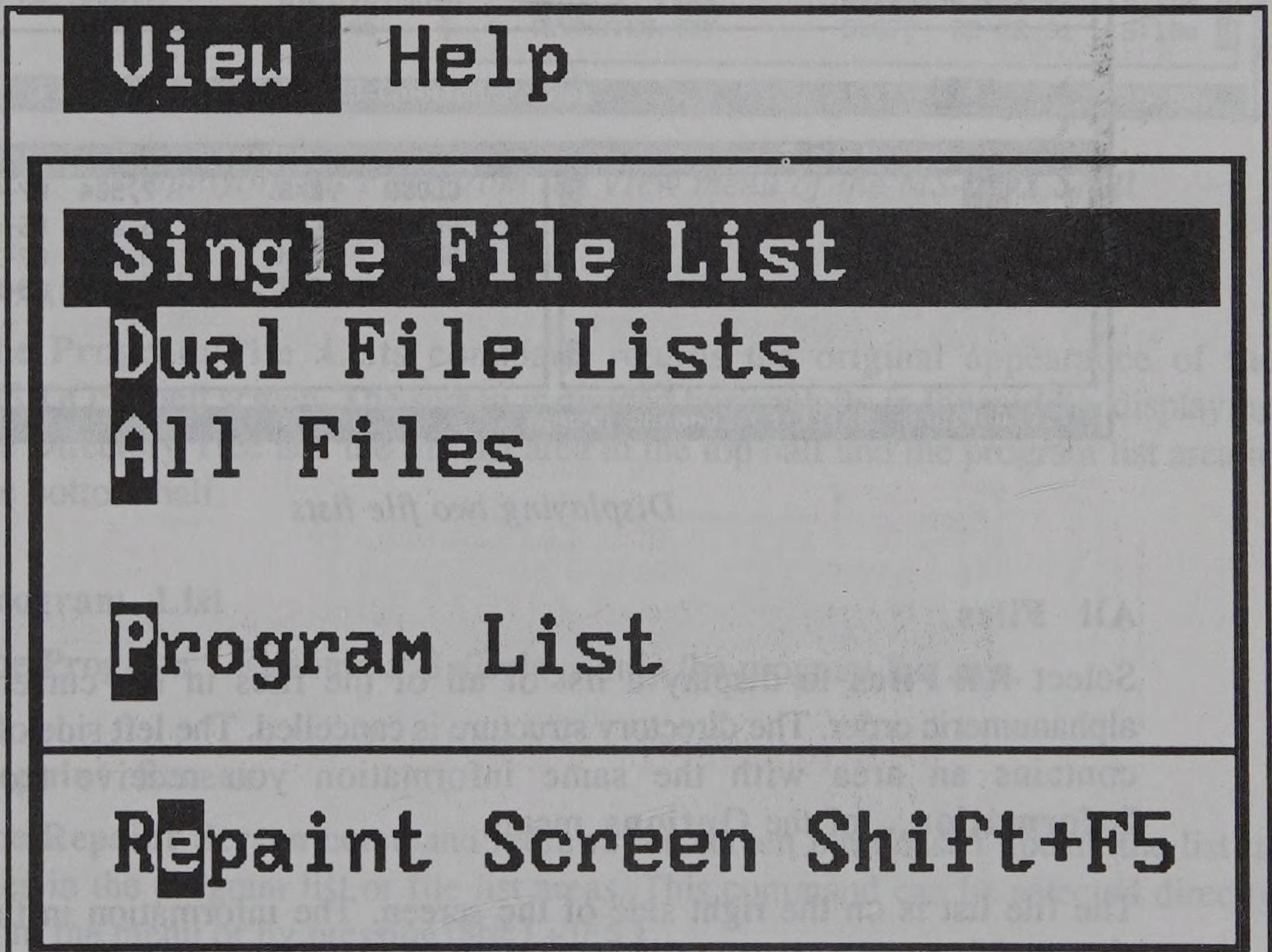
Selecting the color display using Colors from the Options menu

If you don't like the colors that are used, design one according to your preferences. To design your own color scheme, you must edit the DOSSHELL.INI file, in which the color palettes are defined.

Never edit the DOSSHELL.INI file while the MS-DOS Shell is running. The MS-DOS Shell notes the position of this file's cluster on the storage medium (diskette, hard drive) when it starts and immediately writes all changes to this cluster. Making changes while the MS-DOS Shell is running could lead to data loss and crash the Shell.

15.5.4 The View menu

This menu is used to change the display mode of the MS-DOS Shell. You can choose to display only the program list area or only the Directory Tree/file list area on the screen. You can also display both areas on the screen simultaneously. The **View** menu also has a command for dividing the Directory Tree/file list area into one or two file lists and listing all of the files of a drive in alphanumeric order.



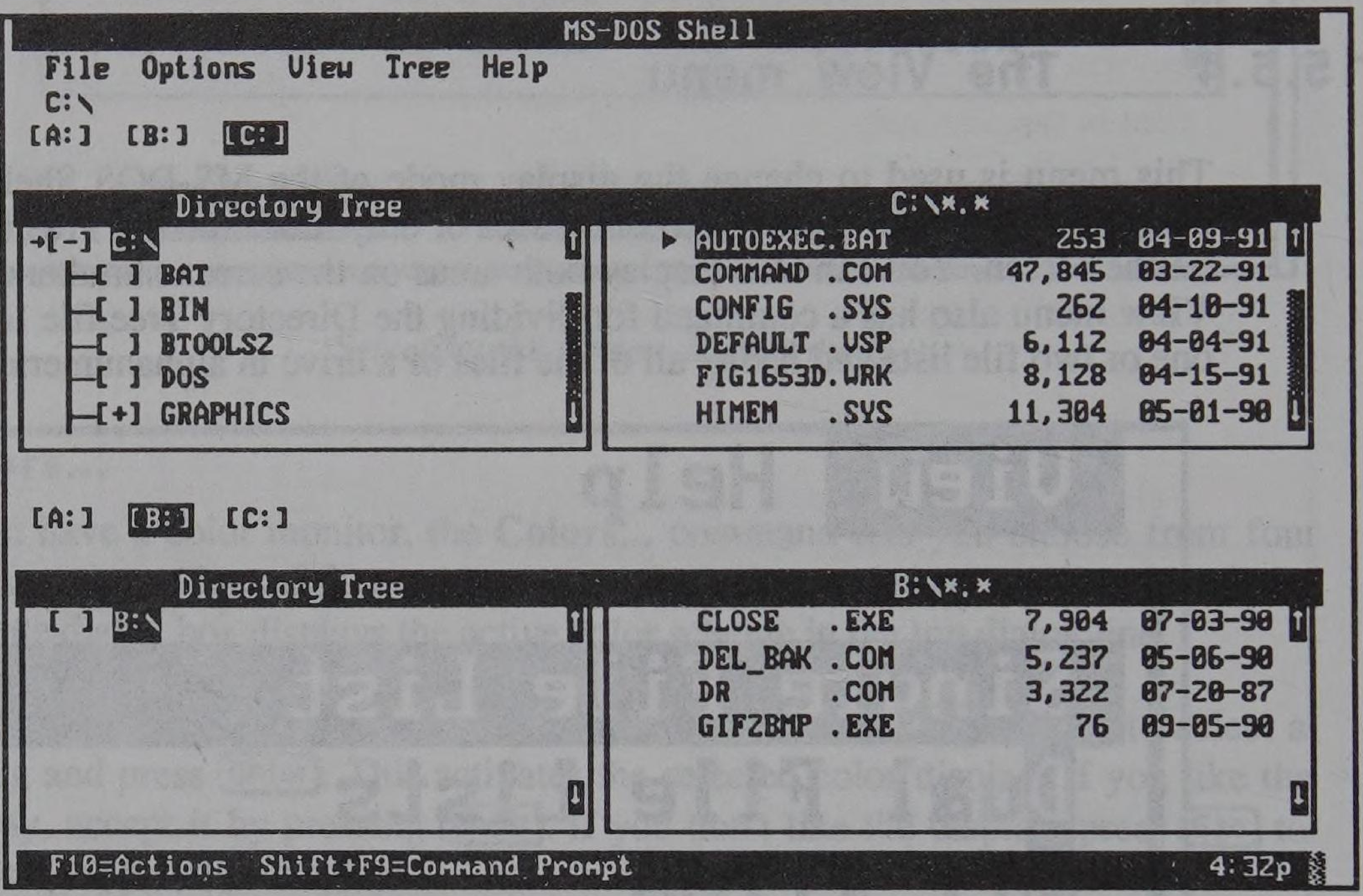
The View menu

Single File List

Select **Single File List** and the Directory Tree/file list area fills the entire screen with a single file list. The left side of the screen displays the Directory Tree while the right side of the screen displays the file list of the selected directory.

Dual File Lists

Dual File Lists divides the Directory Tree/file list area horizontally on the screen. This lets you see the directory structure for two drives on the left side of the screen and the file list of the current directory on the right side of the screen. Use the drive icons to select which drive and file list is displayed. These lists can be two different drives or directories on the same drive.



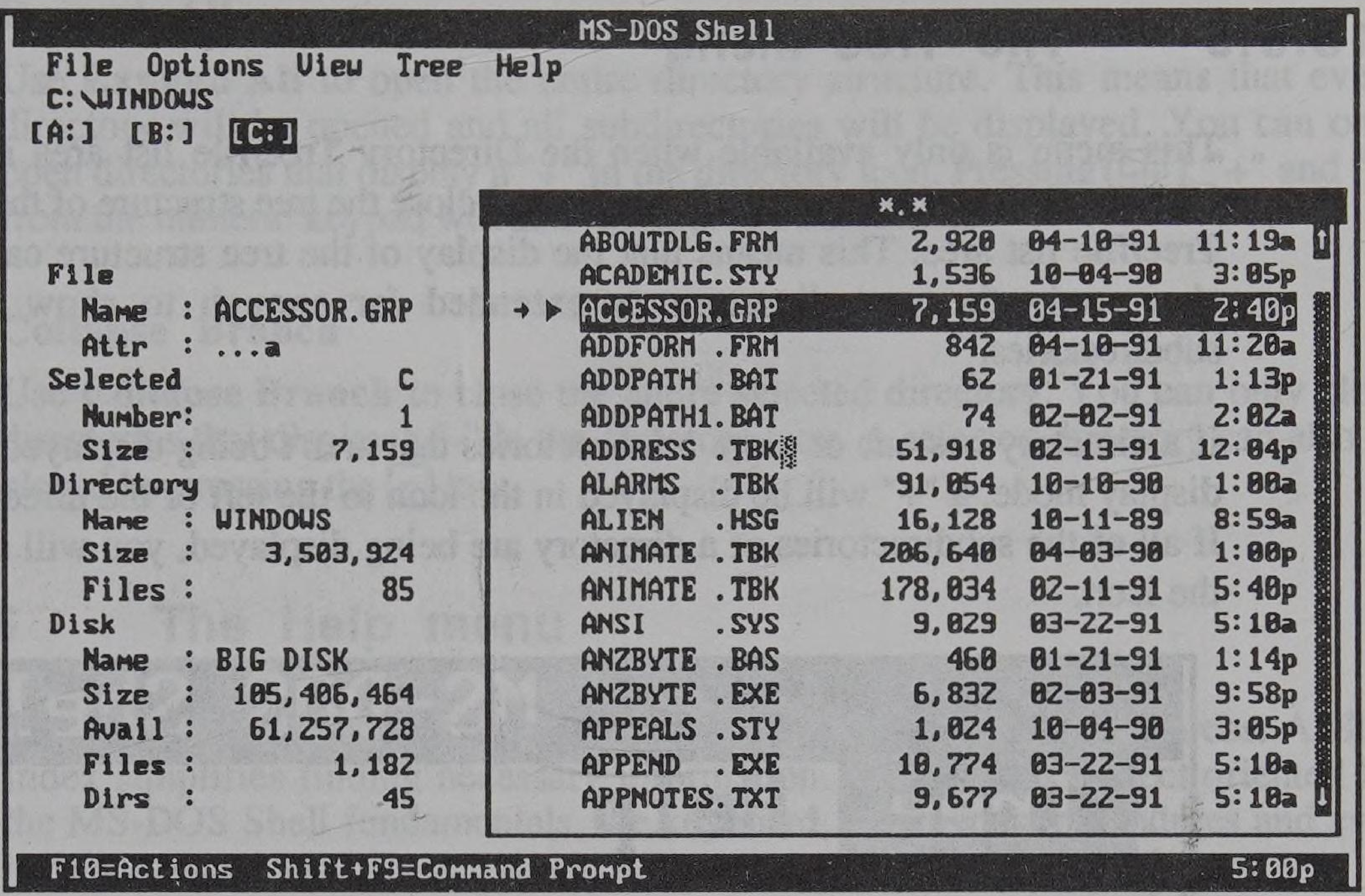
```
                                   MS-DOS Shell
 File  Options  View  Tree  Help
 C:\
[A:]  [B:]  [C:]
          Directory Tree                              C:\*.*
→[-] C:\                           ↑  ▶ AUTOEXEC.BAT       253  04-09-91 ↑
    ├[ ] BAT                             COMMAND .COM    47,845  03-22-91
    ├[ ] BIN                             CONFIG  .SYS       262  04-10-91
    ├[ ] BTOOLS2                         DEFAULT .VSP     6,112  04-04-91
    ├[ ] DOS                             FIG1653D.WRK     8,128  04-15-91
    ├[+] GRAPHICS                      ↓  HIMEM   .SYS    11,304  05-01-90 ↓

[A:]  [B:]  [C:]
          Directory Tree                              B:\*.*
 [ ] B:\                           ↑     CLOSE   .EXE     7,904  07-03-90 ↑
                                         DEL_BAK .COM     5,237  05-06-90
                                         DR      .COM     3,322  07-20-87
                                         GIF2BMP .EXE        76  09-05-90
                                   ↓                                     ↓
 F10=Actions  Shift+F9=Command Prompt                                4:32p
```

Displaying two file lists

All Files

Select **All Files** to display a list of all of the files in the current drive in alphanumeric order. The directory structure is cancelled. The left side of the screen contains an area with the same information you receive from **Show Information...** of the **Options** menu.

The file list is on the right side of the screen. The information in the left area always refers to the selected file in the right area.



```
                                   MS-DOS Shell
 File  Options  View  Tree  Help
   C:\WINDOWS
 [A:]  [B:]  [C:]
                                                   *.*
                              ABOUTDLG.FRM       2,920  04-10-91  11:19a
 File                         ACADEMIC.STY       1,536  10-04-90   3:05p
   Name  : ACCESSOR.GRP   → ▶ ACCESSOR.GRP       7,159  04-15-91   2:40p
   Attr  : ...a               ADDFORM .FRM         842  04-10-91  11:20a
 Selected          C          ADDPATH .BAT          62  01-21-91   1:13p
   Number:          1         ADDPATH1.BAT          74  02-02-91   2:02a
   Size  :      7,159         ADDRESS .TBK      51,918  02-13-91  12:04p
 Directory                    ALARMS  .TBK      91,054  10-10-90   1:00a
   Name  : WINDOWS            ALIEN   .HSG      16,128  10-11-89   8:59a
   Size  :   3,503,924        ANIMATE .TBK     206,640  04-03-90   1:00p
   Files :         85         ANIMATE .TBK     178,034  02-11-91   5:40p
 Disk                         ANSI    .SYS       9,029  03-22-91   5:10a
   Name  : BIG DISK           ANZBYTE .BAS         460  01-21-91   1:14p
   Size  : 105,406,464        ANZBYTE .EXE       6,832  02-03-91   9:58p
   Avail :  61,257,728        APPEALS .STY       1,024  10-04-90   3:05p
   Files :      1,182         APPEND  .EXE      10,774  03-22-91   5:10a
   Dirs  :         45         APPNOTES.TXT       9,677  03-22-91   5:10a
 F10=Actions  Shift+F9=Command Prompt                              5:00p
```

Displaying All Files from the View menu of the MS-DOS Shell

Program/File Lists

The **Program/File Lists** command returns the original appearance of the MS-DOS Shell screen. The screen is divided horizontally in the middle, displaying the Directory Tree and the file list area in the top half and the program list area in the bottom half.

Program List

The **Program List** command displays only the program list area.

Repaint Screen

The **Repaint Screen** command redraws the screen but doesn't update the list of files in the program list or file list areas. This command can be selected directly from the menu or by pressing Shift + F5.

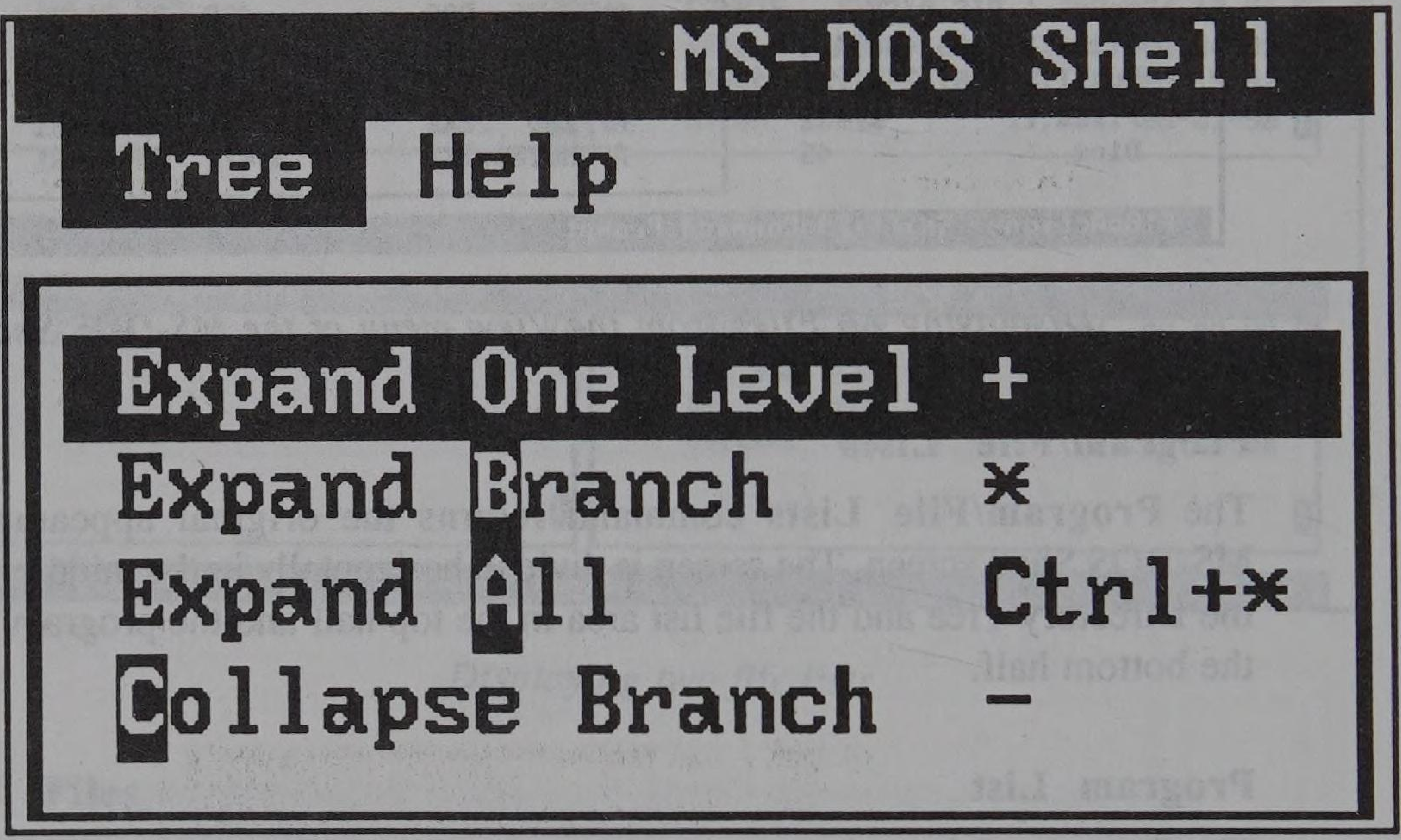
Refresh

Refresh rereads the disk and updates the program list area and the file list area to reflect any changes, such as deleted or restored files. Select **Refresh** either from the menu or by pressing the F5 key.

15.5.5 The Tree menu

This menu is only available when the Directory Tree/file list area is selected. Commands found here are used to open and close the tree structure of the Directory Tree/file list area. This means that the display of the tree structure can be set to show only the root directory or extended far enough to show all of the subdirectories.

If a directory has one or more subdirectories that aren't being displayed due to the display mode, a "+" will be displayed in the icon to the left of the directory name. If all of the subdirectories of a directory are being displayed, you will see a "-" in the icon.



The Tree menu of the MS-DOS Shell

Expand One Level +
Choose **Expand One Level** to open the selected directory by one level. If this directory has subdirectories, they remain closed. You can only open directories that display a "+" in the directory icon. Pressing the + key also opens a selected directory.

Expand Branch

Select **Expand Branch** to open the entire selected directory. If this directory has subdirectories containing other subdirectories, all of the directories are displayed. You can only open directories that display a "+" in the directory icon. The branches can also be expanded by pressing the "*" key.

Expand All

Use **Expand All** to open the entire directory structure. This means that every directory will be opened and all subdirectories will be displayed. You can only open directories that display a "+" in the directory icon. Pressing Ctrl, "+" and "*" from the numeric keypad will also execute this command.

Collapse Branch

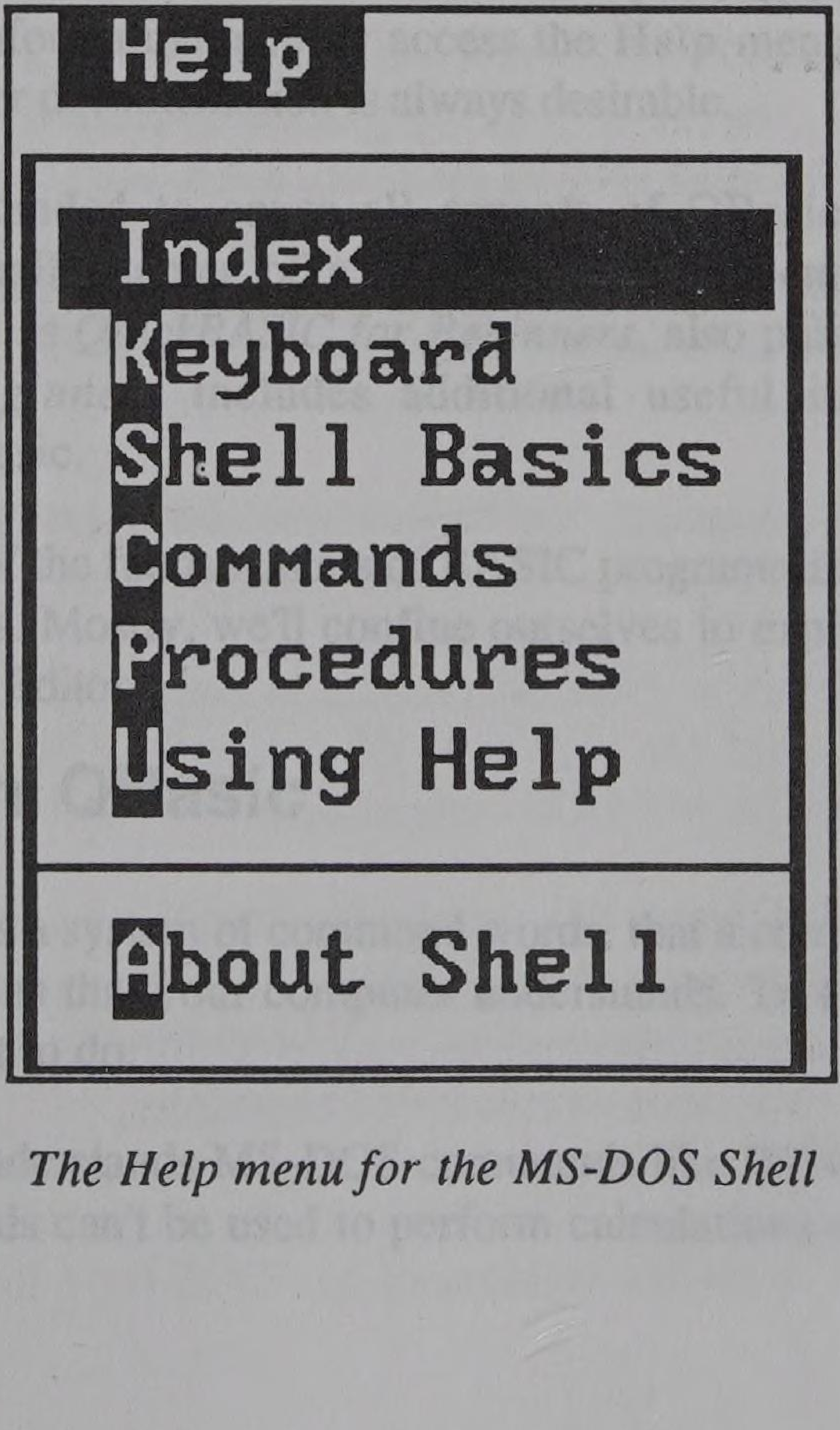
Use **Collapse Branch** to close the entire selected directory. You can only close directories that display a "-" in the directory icon. A selected directory can also be closed by pressing the - key.

15.5.6 The Help menu

The **Help** menu lists the commands arranged according to subject areas. A Help Index simplifies finding necessary information. Help is also subject oriented for the MS-DOS Shell fundamentals, the keyboard, commands, procedures and even help on using Help.

If an item of the **Help** menu is more than one page long, the contents can be scrolled. Either click the arrows of the scroll bar with your mouse or use ↑ and ↓ or PgUp and PgDn keys to scan the help screens.

Press Esc to exit the **Help** menu.



The Help menu for the MS-DOS Shell

Index

Select **Index** to view a help screen listing subjects from A to Z.

Keyboard

Select **Keyboard** to learn about the most important keys and key combinations.

Shell Basics

Select **Shell Basics** to find help about MS-DOS Shell fundamentals.

Commands

Select **Commands** to find help about MS-DOS Shell commands.

Procedures

Select **Procedures** to find help about MS-DOS Shell procedures.

Using Help

Select **Using Help** to learn about using the help system.

About Shell

Select **About Shell** to display the MS-DOS Shell version number and the copyright notice.

Chapter 16

Using QBasic

Some PC users are content to use only commercial applications. Others, however, are more open to the possibilities that programming languages offer for writing their own applications. QBasic is one of these programming languages and is part of MS-DOS 5.0. This new language replaces the GW-BASIC language that was part of the earlier versions of DOS.

QBasic includes an extensive set of online documentation. You'll be able to learn about using the various commands and functions by using these help screens. Whenever you need information, simply access the **Help** menu from the QBasic Editor. However, further documentation is always desirable.

This chapter isn't intended to cover all aspects of QBasic. That would be impossible in one small chapter of a book. We recommend obtaining other reference sources such as *QuickBASIC for Beginners*, also published by Abacus. *QuickBASIC for Beginners* includes additional useful information about programming with QBasic.

We'll introduce a few of the fundamentals of BASIC programming and create some small sample programs. Mostly, we'll confine ourselves to explaining how to use QBasic and the QBasic Editor.

16.1 Working With QBasic

A computer language is a system of command words, that a compiler or interpreter changes into instructions that your computer understands. To be more precise: It tells the computer what to do.

Your computer also understands MS-DOS commands like DISKCOPY or TYPE, but MS-DOS commands can't be used to perform calculations or create graphics.

The BASIC programming language has commands for performing these and other operations. Best of all, BASIC is easy to learn.

Back when computers were still gigantic machines and very expensive, only specialists could communicate with them. Even today computers really understand only one language, machine language. On top of that, this language varies from one type of computer to another.

As computers became smaller and more reasonably priced, it wasn't practical for every user to learn how to write complicated machine language programs to communicate with their computers. Languages were developed that were easy to learn. One such language was BASIC.

How does BASIC communicate with a computer?

BASIC uses a set of defined commands, much like the ones used in MS-DOS. These commands are linked together in such a way that the computer solves problems for us, step by step. Each problem presented to the computer must be divided into small steps, which are combined into programs. A program is a series of steps for solving a problem.

16.1.1 Special features of QBasic

Computers are all different inside. They have different amounts of RAM (random access memory) and even use different microprocessors. A microprocessor is the central control unit of your computer. MS-DOS computers today use microprocessors such as the 8086, 8088, 80286, 80386 and 80486.

Computers such as the Apple Macintosh use microprocessors from the 68000 family. These microprocessors are quite different and understand different instructions.

Machine language, the lowest level for communicating with your computer, is very different for each family of microprocessors.

There are even great differences in languages like BASIC. Different versions of BASIC exist for each type of computer. QBasic was developed by Microsoft especially for MS-DOS computers.

Since the different PCs usually conform to a standard, called the industrial standard, QBasic can be used on any MS-DOS computer. That means that you can write programs, then share them with your friends, and - more importantly - you can also use programs written by other people.

QBasic is a further development of the programming language, GW-BASIC, which was part of earlier versions of MS-DOS for a long time. QBasic is a lot easier to use and more efficient. Beginners will find helpful menus making it much easier to use.

QBasic also uses an intelligent editor, checking each line for proper syntax and providing hints on correcting the error.

The QBasic Editor offers an easy and convenient way to enter a program since it is a full-screen editor similar to a word processor. This makes QBasic a lot handier for creating large programs.

Although you can still use line numbers (for reasons of compatibility with other BASIC language programs), they are no longer necessary in QBasic. QBasic can use programs created with GW-BASIC language. This makes it easier to switch over to using QBasic.

If you want to convert your QBasic programs to stand alone programs which don't require the QBasic interpreter to run, you can buy Microsoft's QuickBASIC. Then you can transform the completed programs into an EXE program which runs independently. QuickBASIC is more powerful than QBasic and is compatible with QBasic.

16.1.2 Starting QBasic

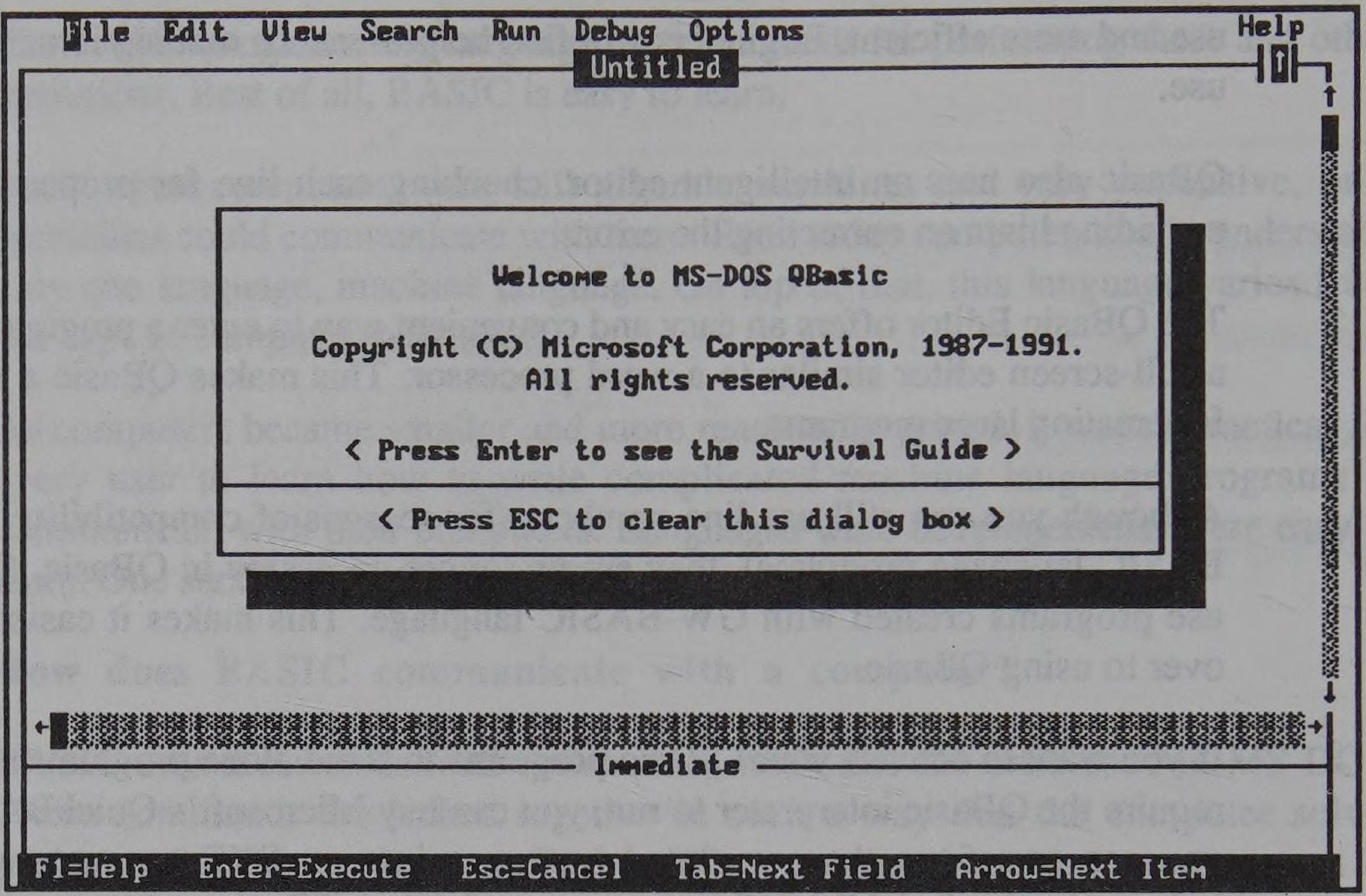
To work with QBasic, you must first install the program. If you installed all files provided with MS-DOS 5.0 on the hard drive and the DOS directory is in the search path, you can start the QBasic program easily.

If QBasic is not in a directory defined by your search path (as explained previously) you must make the directory containing QBasic the current directory. It's a good idea to put QBasic and the programs created with it in a separate directory, such as \QBASIC.

To start QBasic from the system prompt in the command interpreter, type the program name:

```
QBASIC
```

The startup screen of QBasic will appear:


File Edit View Search Run Debug Options Help
Untitled
Immediate
F1=Help Enter=Execute Esc=Cancel Tab=Next Field Arrow=Next Item

Welcome to MS-DOS QBasic

Copyright (C) Microsoft Corporation, 1987-1991.
All rights reserved.

< Press Enter to see the Survival Guide >

< Press ESC to clear this dialog box >

The QBasic startup screen

A dialog box appears in the middle of the screen with a note about online help information. Press Esc to clear this dialog box from the screen.

The QBasic startup screen

The first line of the screen is the menu bar. You can use the menus to choose the QBasic operations and processes.

The next line is the area title and displays the name of the program currently in the memory. If a program has not been loaded yet, the line displays "Untitled". This is the default screen you always see when starting QBasic without entering any optional parameters.

The actual work area takes up the greatest amount of space on the screen.

Beneath the work area, is an area titled "Immediate". You can use this area to enter one line commands directly. The computer executes these commands as soon as Enter is pressed. Press F6 to make this area current. Pressing F6 again will return to the work area.

The last line on the screen is the status bar. It displays information about the selected menu. If no menu has been selected, the commands displayed in the status bar can be selected by pressing the appropriate function key, or with your mouse.

The lower right corner displays the line and column number of where your cursor is located. This is practical, especially when working on lengthy programs.

16.1.3 Running QBasic

QBasic allows you to:

- Enter a program just as you would text in a word processor. As you enter a program, QBasic operates much like EDIT, the MS-DOS Editor.
- Select operations from the menus at the top of the screen. You can load and save finished programs as files, start the current program, etc.
- Use QBasic commands to write and modify a program.

The QBasic Editor

The QBasic Editor is a full screen editor that lets you create, view and modify program files quickly and easily. It is similar in many ways to the MS-DOS Editor. You can view a file on the entire screen and use the cursor keys to easily change the contents of the file.

Moving the cursor in the program editor

When QBasic is first started, the cursor is located in the upper left corner of the work area. You are ready to begin making entries in QBasic. The following list summarizes the keys you can use to move the cursor:

QBasic Cursor Control Keys

Key	Function
↑	Move one line up
←	Move one character to the left
→	Move one character to the right
↓	Move one line down
PgUp	Move one page up
PgDn	Move one page down
Home	Jump to beginning of line
End	Jump to end of line

Ctrl + ←	Move one word to the left
Ctrl + →	Move one word to the right
Ctrl + PgUp	Move the cursor 78 characters to the left
Ctrl + Home	Jump to beginning of program
Ctrl + End	Jump to end of program
Ctrl + PgDn	Move the cursor 78 characters to the right

Editing programs

Simple corrections to programs are made by deleting the mistake and inserting the correction (in Insert mode) or by typing the change over the error (in Overtype mode). To switch between Insert and Overtype modes, press the Ins key.

You can tell which mode you are in by observing the cursor. If the cursor is a line beneath the characters, QBasic is in Insert mode. If the cursor appears as a small rectangle, then QBasic is in Overtype mode.

Using menus

You can select the QBasic menus by pressing the Alt key to activate the menu bar, then pressing the highlighted letter's key to open the menu. Select a command with the cursor keys and press Enter.

It's even easier to select menus with the mouse. Point to the desired command and click with the left mouse button.

If a command has three periods after the name "...", the command is not executed when you press Enter. Instead, a dialog box appears where you enter additional information. Pressing Esc will cancel a menu selection or input to a dialog box.

The pull-down menus

The pull-down menus are lists of commands. For each command, one letter is highlighted. To select a command, press the desired highlighted letter. You can also use the ↑ and ↓ cursor keys to select a command. When the selection cursor covers the desired command, press Enter to select it.

The QBasic scroll bar

On the right margin of the screen is a vertical strip with arrows pointing up and down. This is the scroll bar. Within this strip is a rectangle. This is the scroll slider. Its position within the scroll bar indicates which section of the file is being displayed.

If the slider is at the top of the scroll bar, the start of the file is being displayed. If it is in the middle of the scroll bar, you are seeing the middle portion and if it's at the bottom, the area contains the last part of the file.

Use the ↑ or ↓ cursor keys to move through the file one line at a time. Press PgUp and PgDn to scroll the entire screen contents up or down by one screen.

You can also click the mouse pointer on either of the two direction arrows at the top and bottom of the scroll bar to scroll the data in the area up or down. You can even move the slider with the mouse.

Place the mouse pointer on the slider, press the left mouse button and, while holding down the mouse button, move the mouse pointer in the desired direction. The scroll slider will move along in the direction of the mouse arrow and display the corresponding contents of the file in the area.

The horizontal scroll bar on the bottom of the screen works exactly like the vertical bar, except that it moves horizontally. Use the key combinations Ctrl + PgDn to move right and Ctrl + PgUp to move left.

16.2 The QBasic Menus

Using the letter keys

Press Alt to activate the menu bar. Then press the highlighted letter key to pull down the individual menus. Pressing the highlighted letter opens the menu.

The menu bar displays the titles for each of the menus. These titles are **File**, **Edit**, **View**, **Search**, **Run**, **Debug**, **Options** and **Help**. Various commands are available only at specific times and some commands open dialog boxes for further input.

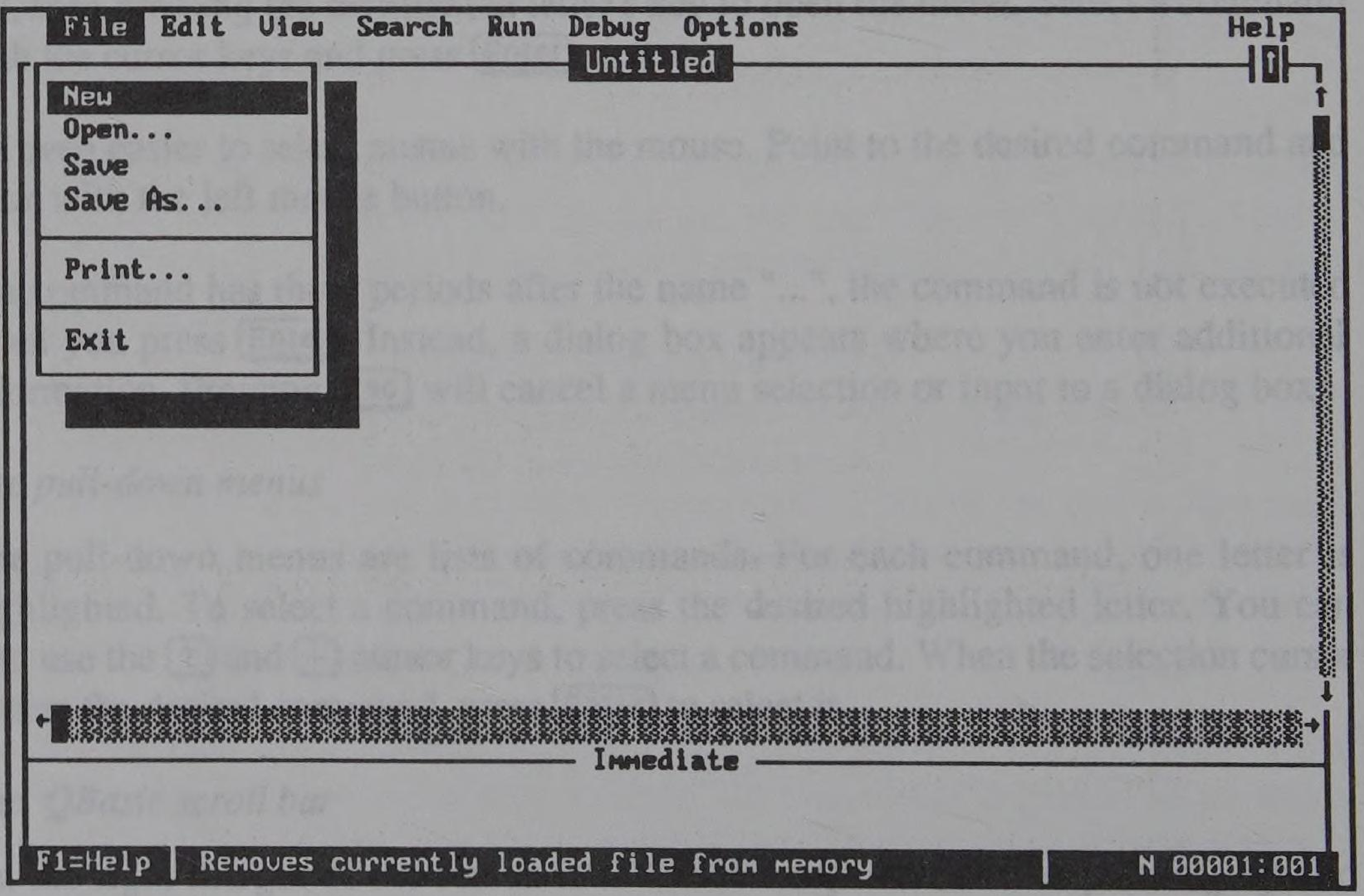
Using the cursor keys

The ↓ and ↑ cursor keys can also be used to open the pull-down menus, while the → and ← cursor keys can be used to select a pull-down menu. Press one of the cursor keys to highlight the desired command and press Enter to execute the command.

Using the mouse

If you are using the mouse, point and click to the desired menu and the pull-down menu will open. You can also activate individual commands by clicking on them with the mouse.

Here's how the menus are organized and what the commands do.



```
F1=Help │ Removes currently loaded file from memory          │   N 00001:001
```

The QBasic File menu

The File menu

New Clears the screen and deletes the file contents.

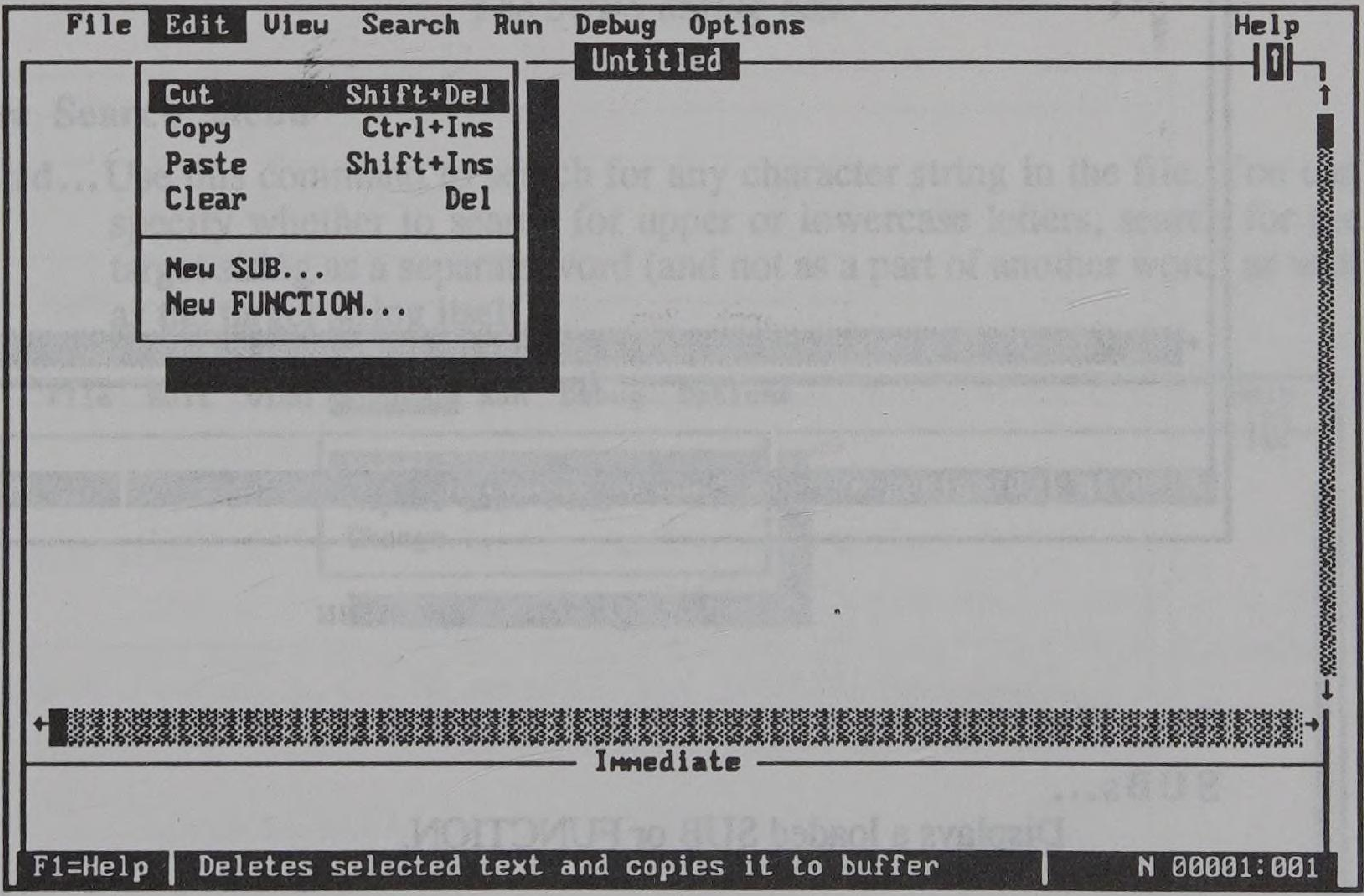
Open...
Loads an existing file from a diskette or hard drive to command memory.

Save
Saves a file which has been previously saved. If you haven't saved a file yet, this command prompts you to enter a filename.

Save As...
Saves a new file, or saves an existing file with a new name.

Print...
Prints the entire file or a marked block of text on the printer.

Exit Leaves QBasic and returns to the MS-DOS Shell or command line.


File
Edit
View
Search
Run
Debug
Options
Help
Untitled
Cut Shift+Del
Copy Ctrl+Ins
Paste Shift+Ins
Clear Del
New SUB...
New FUNCTION...
Immediate
F1=Help

```
Deletes selected text and copies it to buffer
N 00001:001
```

The QBasic Edit menu

The Edit menu

Cut Deletes a marked block of text from the file and copies the block to the buffer.

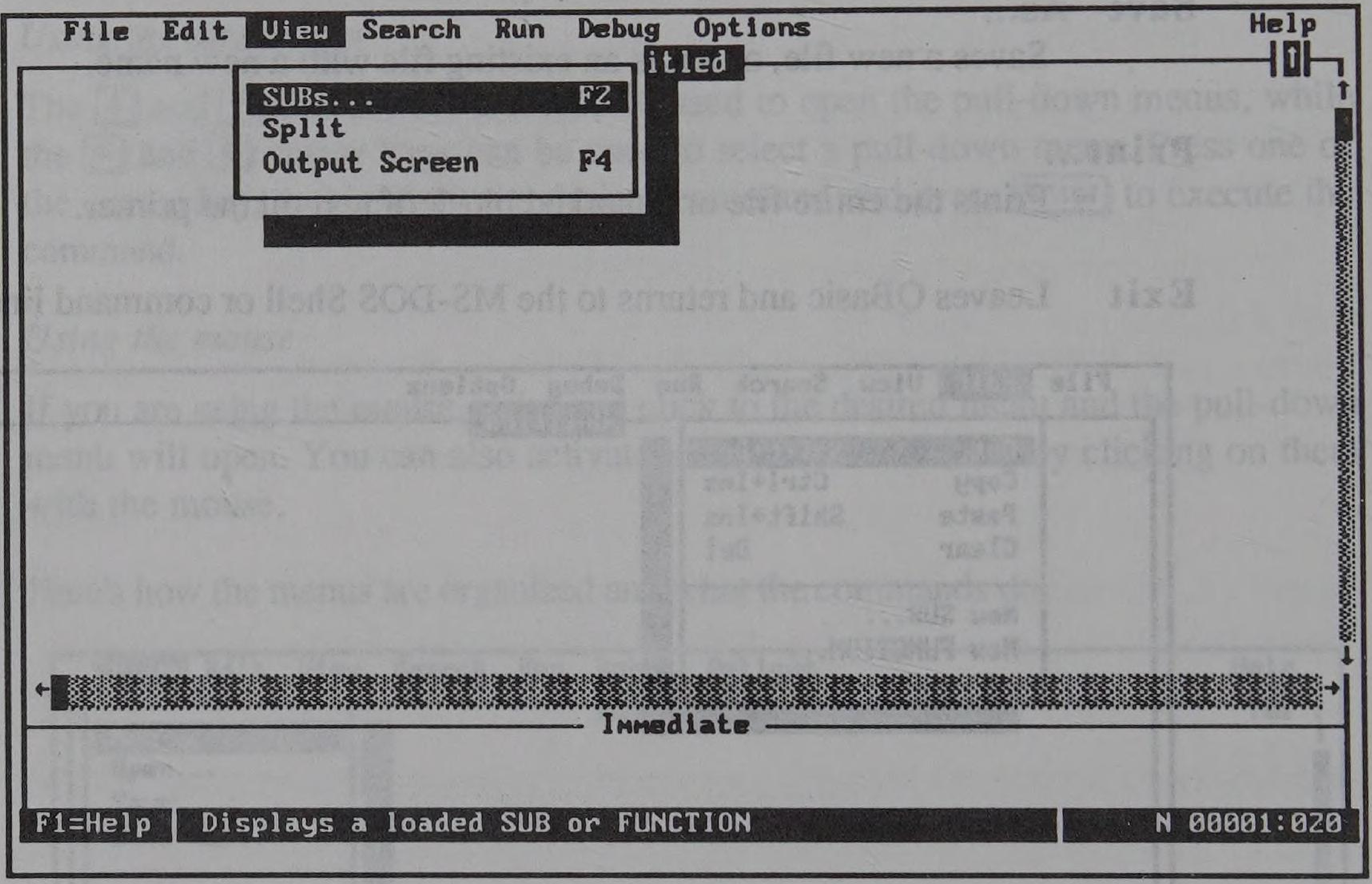
Copy Copies a marked block from the file to the buffer.

Paste Inserts the contents of the buffer (block of text) at the position of the cursor in the file.

Clear Deletes a marked block of text. The deleted text is not copied to the buffer.

New SUB...
Creates a new SUB procedure.

New FUNCTION...
Creates a new FUNCTION procedure.


File
Edit
View
Search
Run
Debug
Options
Help
itled
SUBs... F2
Split
Output Screen F4
Immediate
F1=Help
N 00001:020

```
Displays a loaded SUB or FUNCTION
```

The QBasic View menu

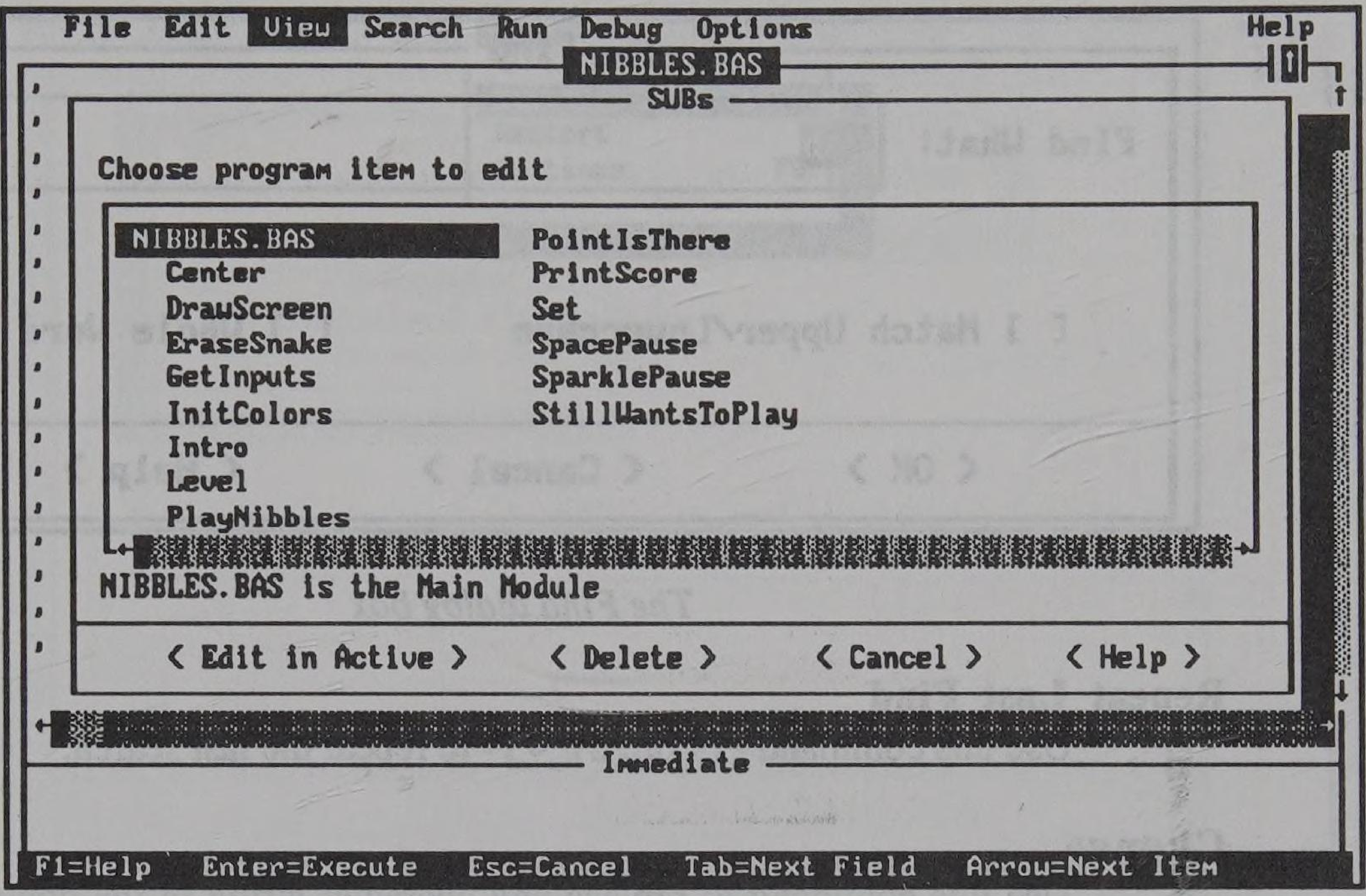
The View menu

SUBs...
Displays a loaded SUB or FUNCTION.

Split Divides the screen into two View windows.

Output Screen
Displays the output screen.

Selecting SUBS with the cursor or the mouse, or pressing F2 displays a dialog box containing the names of SUB and FUNCTION procedures which may be edited or deleted.


File Edit View Search Run Debug Options Help
NIBBLES.BAS
SUBs
< Edit in Active > < Delete > < Cancel > < Help >
Immediate
F1=Help Enter=Execute Esc=Cancel Tab=Next Field Arrow=Next Item

Choose program item to edit

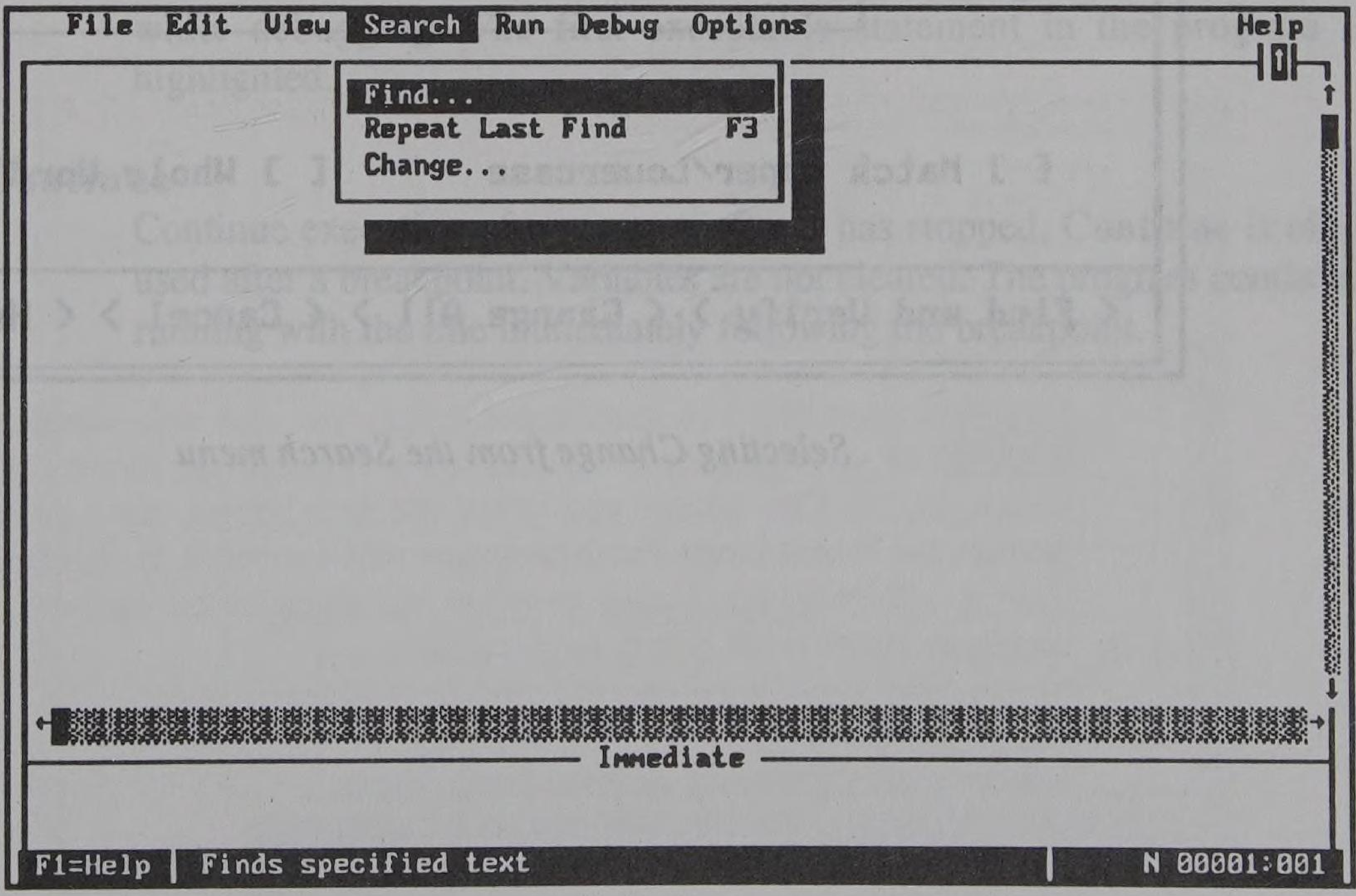
```
NIBBLES.BAS            PointIsThere
    Center             PrintScore
    DrawScreen         Set
    EraseSnake         SpacePause
    GetInputs          SparklePause
    InitColors         StillWantsToPlay
    Intro
    Level
    PlayNibbles
```

NIBBLES.BAS is the Main Module

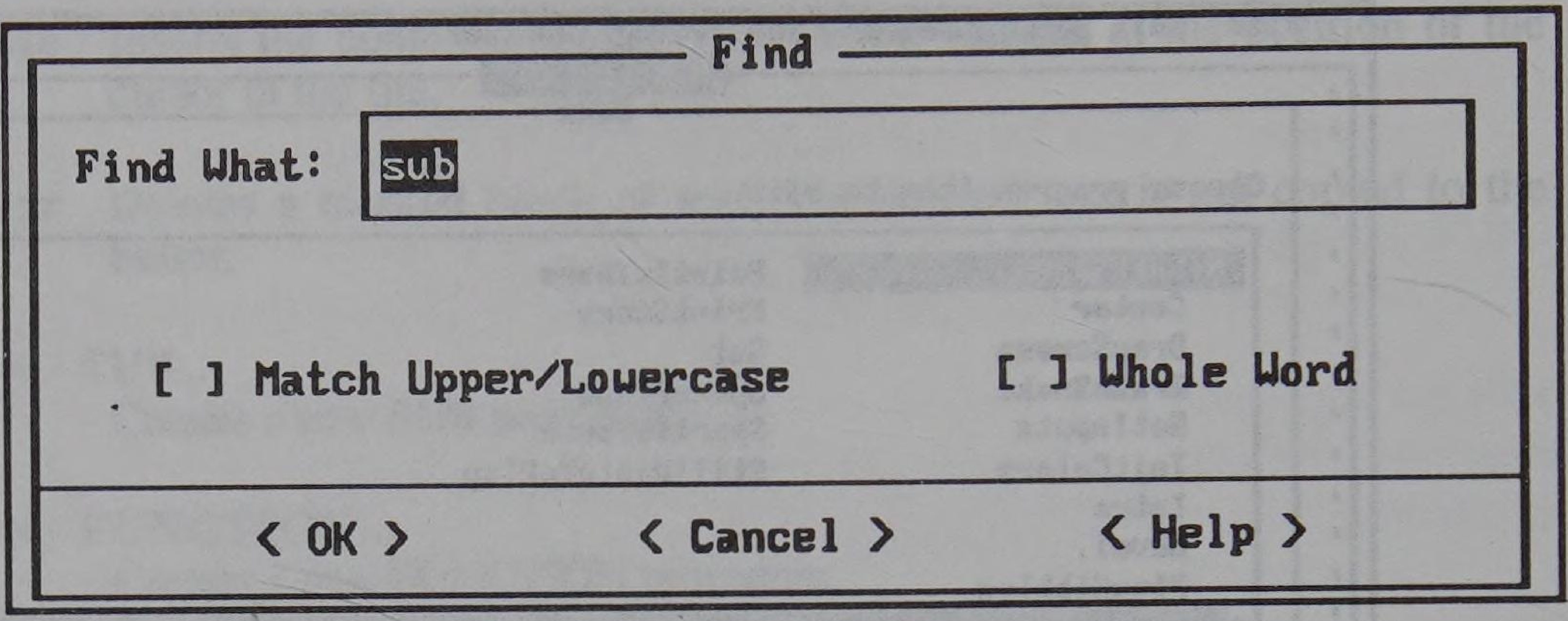
The SUBS dialog box

The Search menu

Find... Use this command to search for any character string in the file. You can specify whether to search for upper or lowercase letters, search for the target string as a separate word (and not as a part of another word) as well as the target string itself.


File Edit View Search Run Debug Options Help
Find...
Repeat Last Find F3
Change...
Immediate
F1=Help
Finds specified text
N 00001:001

The QBasic Search menu


Find
[ ] Match Upper/Lowercase
[ ] Whole Word
< OK >
< Cancel >
< Help >

Find What: sub

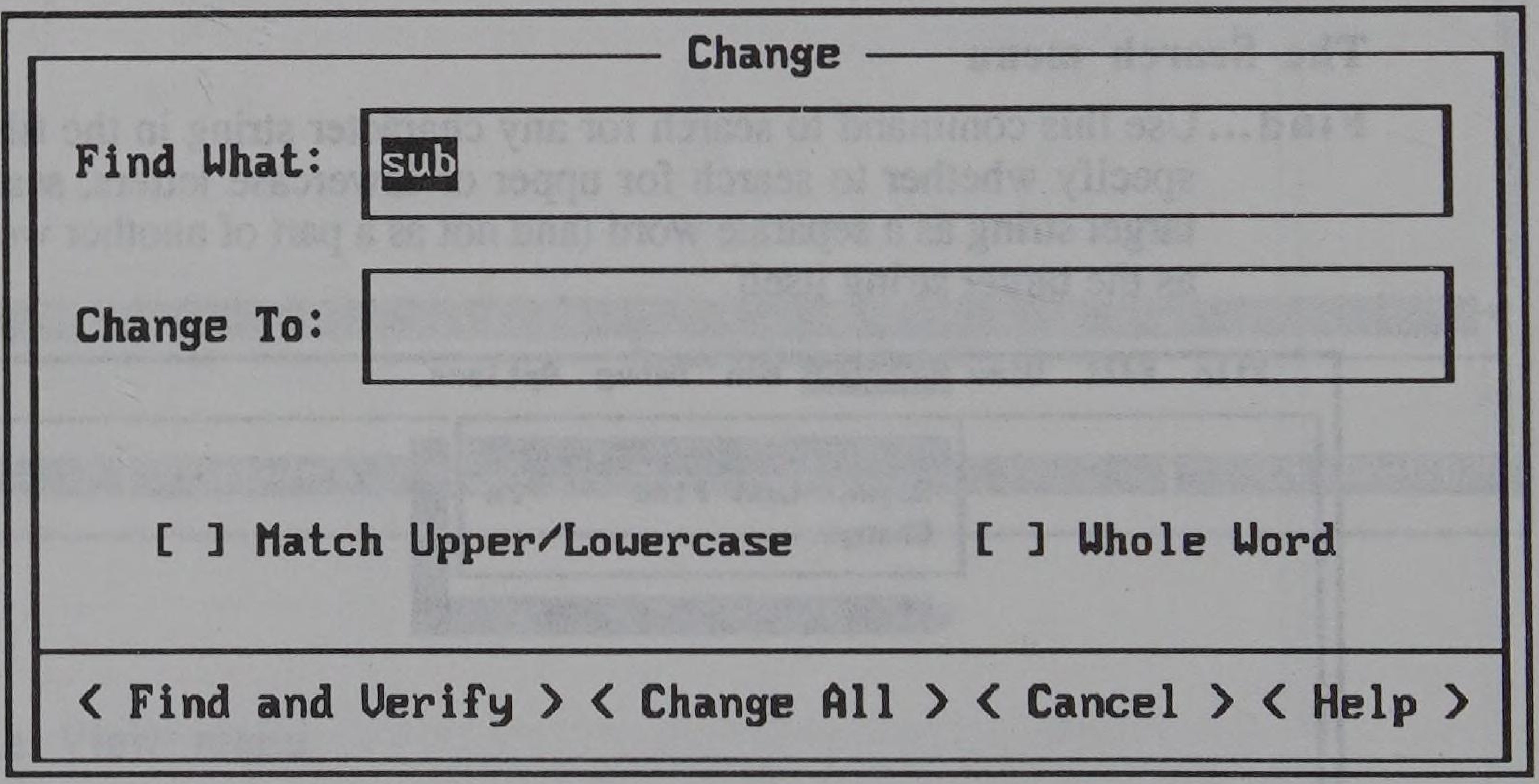
The Find dialog box

Repeat Last Find

Use this command - or press F3 - to repeat the last search.

Change...

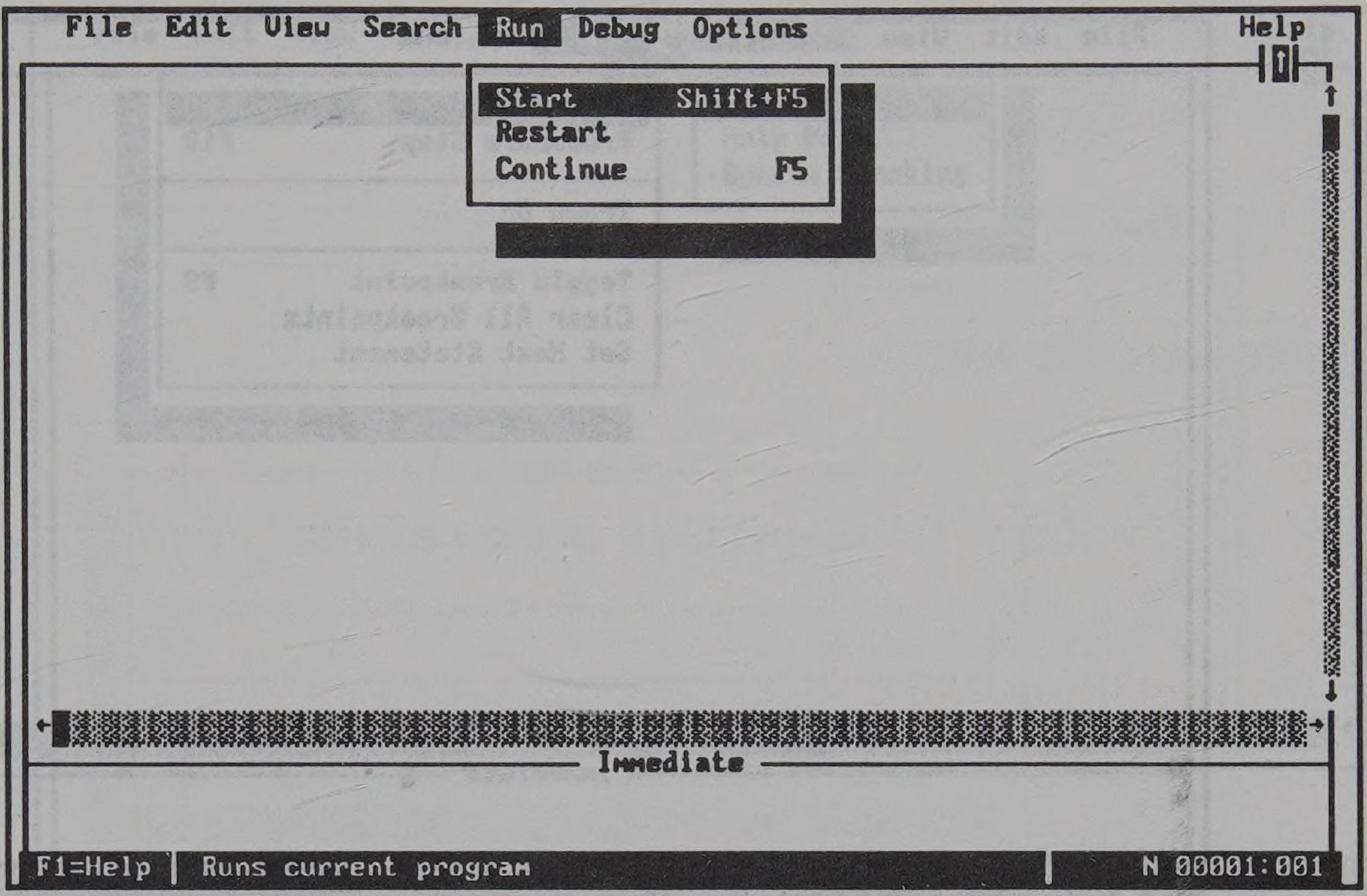
Use this command to replace any character string in the document with any other string of characters.


Change
[ ] Match Upper/Lowercase
[ ] Whole Word
< Find and Verify >
< Change All >
< Cancel >
< Help >

Find What: sub

Change To:

Selecting Change from the Search menu


File
Edit
View
Search
Run
Debug
Options
Help
Start Shift+F5
Restart
Continue F5
Immediate

F1=Help | Runs current program | N 00001:001

The QBasic Run menu

The Run menu

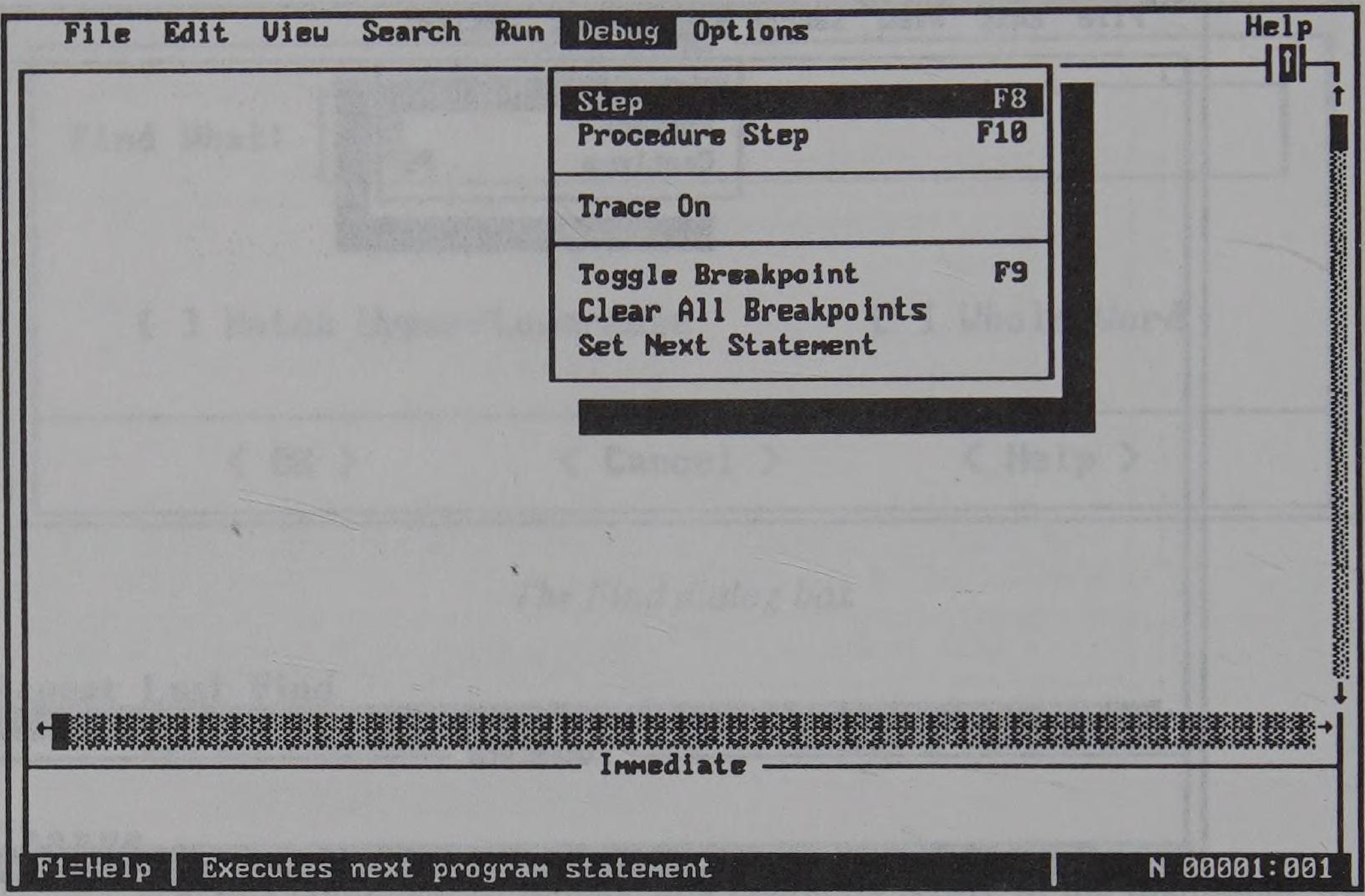
Start Runs program currently loaded in the Editor.

Restart

Clears all variables in preparation for single-stepping through a program while debugging. The first executable statement in the program is highlighted.

Continue

Continue execution of a program after it has stopped. **Continue** is often used after a breakpoint. Variables are not cleared. The program continues running with the line immediately following the breakpoint.


File
Edit
View
Search
Run
Debug
Options
Help
Step F8
Procedure Step F10
Trace On
Toggle Breakpoint F9
Clear All Breakpoints
Set Next Statement
Immediate

```
F1=Help │ Executes next program statement                N 00001:001
```

The QBasic Debug menu

The Debug menu

Step Executes next program statement. This command permits moving through a program one statement at a time while checking for program logic errors.

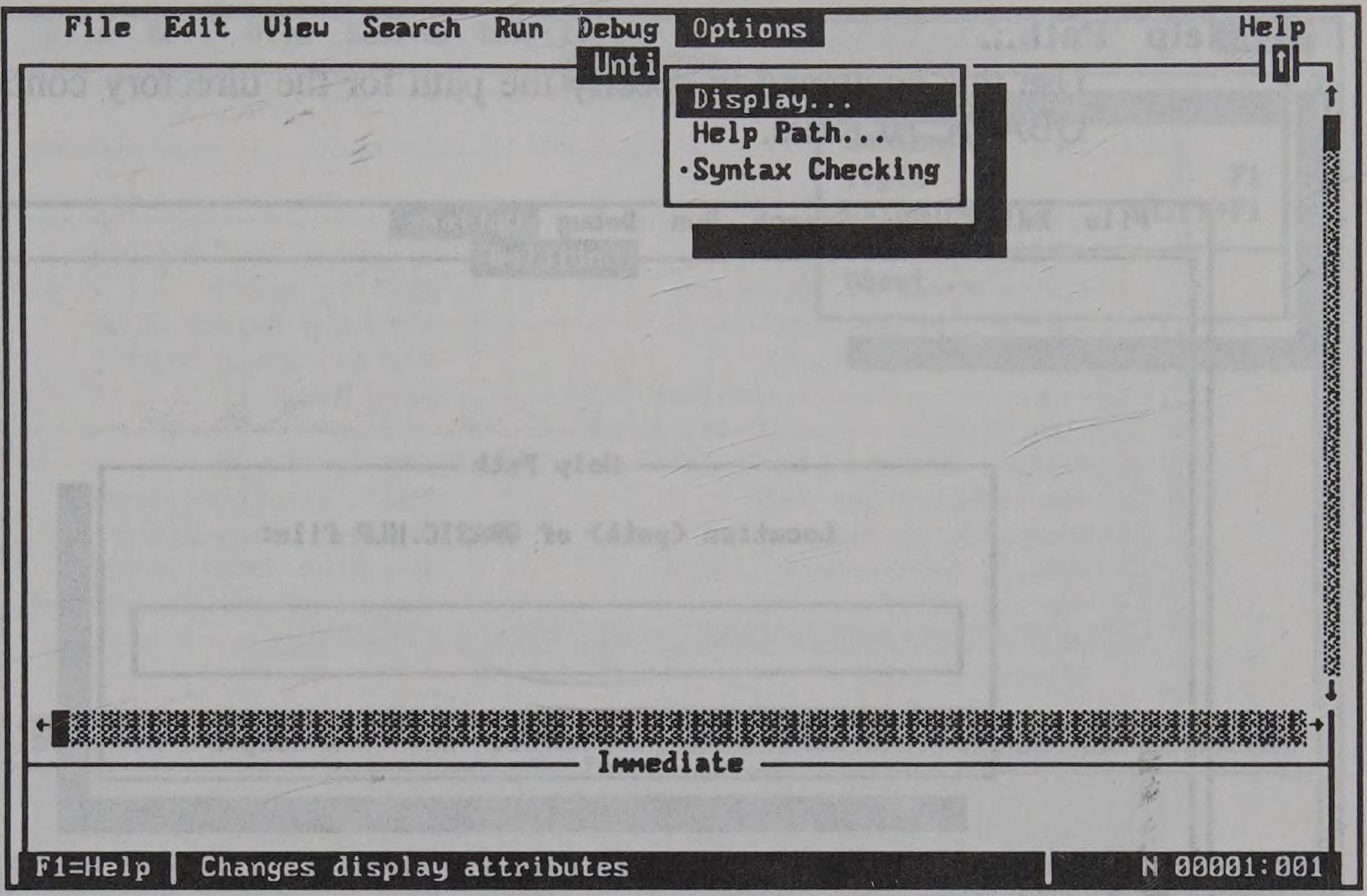
Procedure Step
Single steps through a program, but executes procedure calls as a single statement.

Trace On
Highlights each statement in a program as it is executed. This permits viewing the flow of a program.

Toggle Breakpoint
Use this command to turn breakpoints on and off. Breakpoints are markers in your program. To turn on a breakpoint, move the cursor to highlight the line where you want the breakpoint, then press F9 or choose this command. Your program will run until it reaches this line, then it will stop, permitting printing variables in the Immediate area or single stepping from this point forward.

Clear All Breakpoints
Removes all previously defined breakpoints.

Set Next Statement
Changes the program execution sequence so that the next line executed is the one the cursor is on.


File
Edit
View
Search
Run
Debug
Options
Help
Unti
Display...
Help Path...
·Syntax Checking
Immediate
F1=Help

Changes display attributes

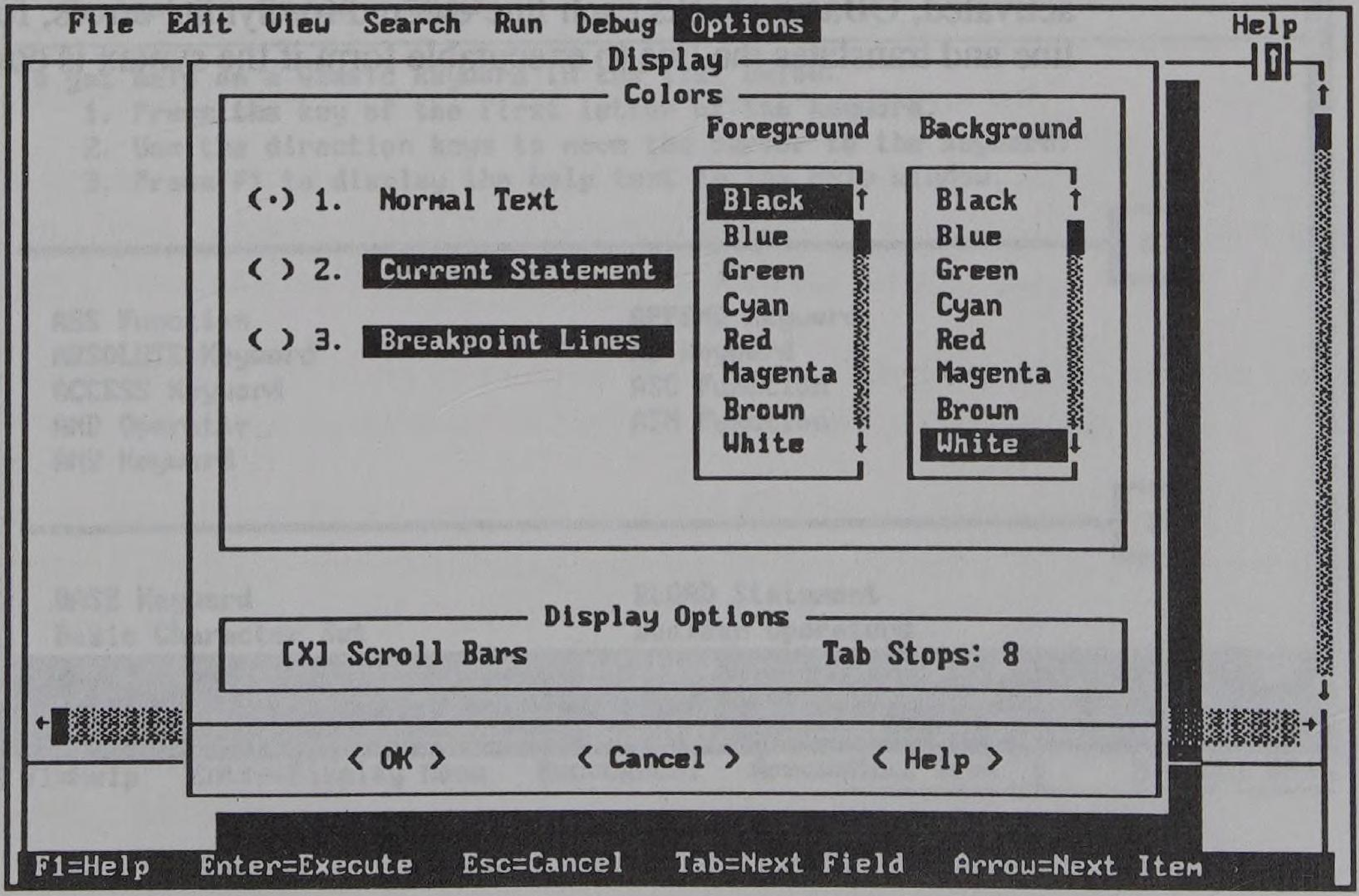
N 00001:001

The QBasic Options menu

The Options menu

Display...

Use this command to select the color of the screen background and the color of the font (the foreground), to switch the scroll bar on and off and to set the distance between the tabs.


File
Edit
View
Search
Run
Debug
Options
Help
Display
Colors
Foreground
Background
(•) 1. Normal Text
( ) 2. Current Statement
( ) 3. Breakpoint Lines
Black
Blue
Green
Cyan
Red
Magenta
Brown
White
Black
Blue
Green
Cyan
Red
Magenta
Brown
White
Display Options

[X] Scroll Bars

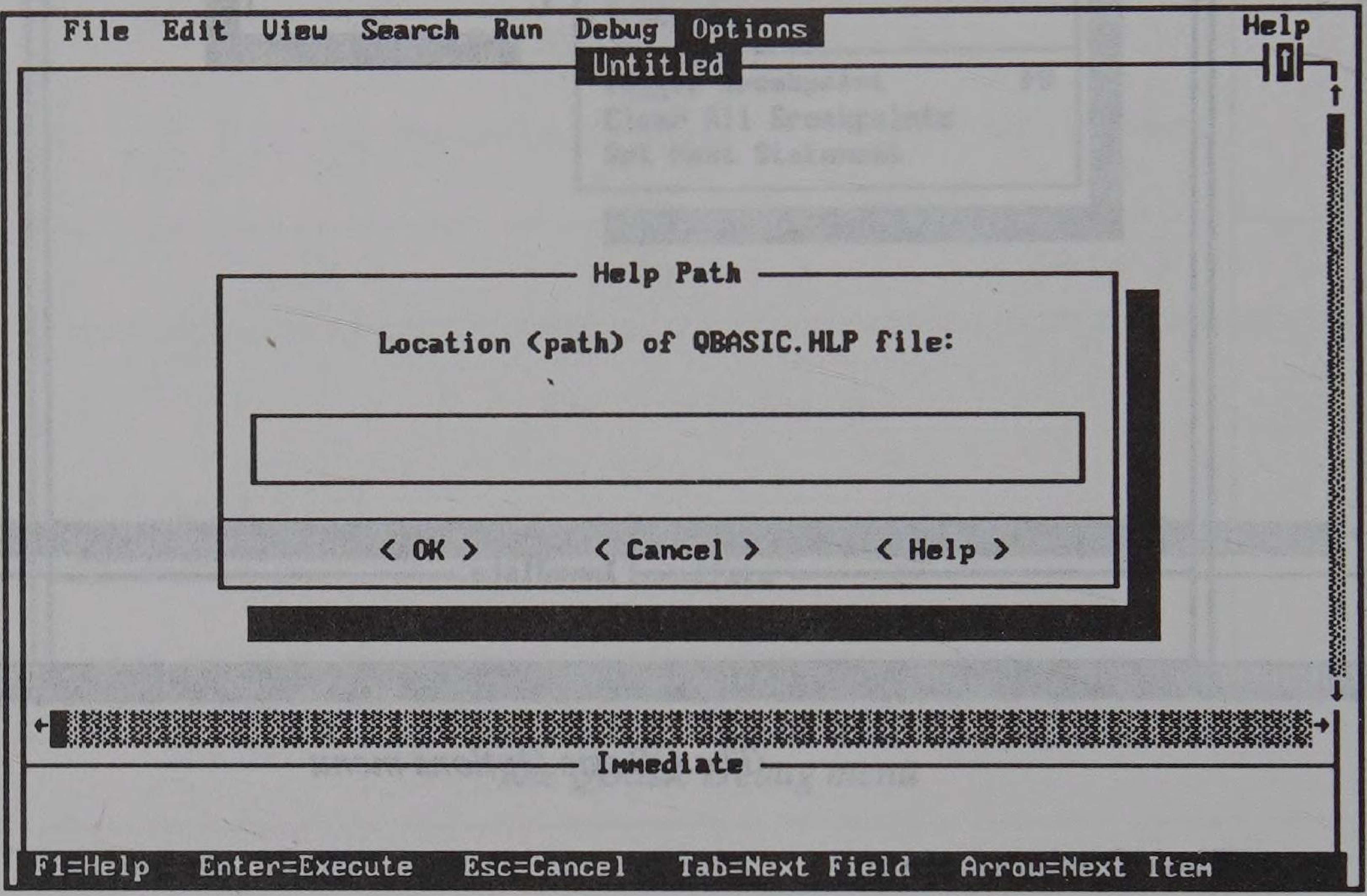
Tab Stops: 8

< OK >
< Cancel >
< Help >
F1=Help
Enter=Execute
Esc=Cancel
Tab=Next Field
Arrow=Next Item

The Display dialog box

Help Path...

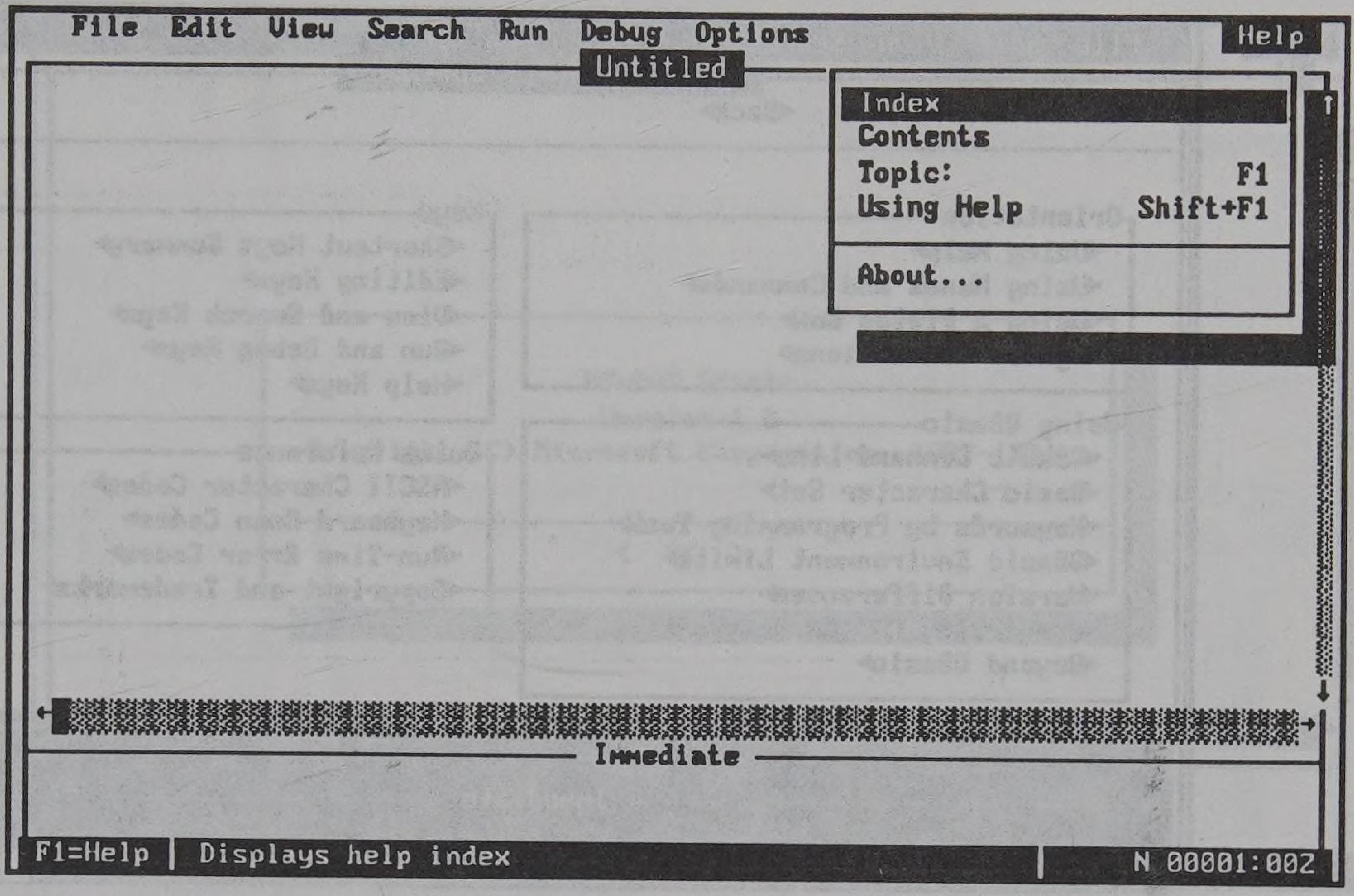
Use this command to specify the path for the directory containing the QBASIC.HLP file.


File Edit View Search Run Debug Options
Help
Untitled
Help Path
Location (path) of QBASIC.HLP file:
< OK >
< Cancel >
< Help >
Immediate
F1=Help Enter=Execute Esc=Cancel Tab=Next Field Arrow=Next Item

The Help Path dialog box

Syntax Checking

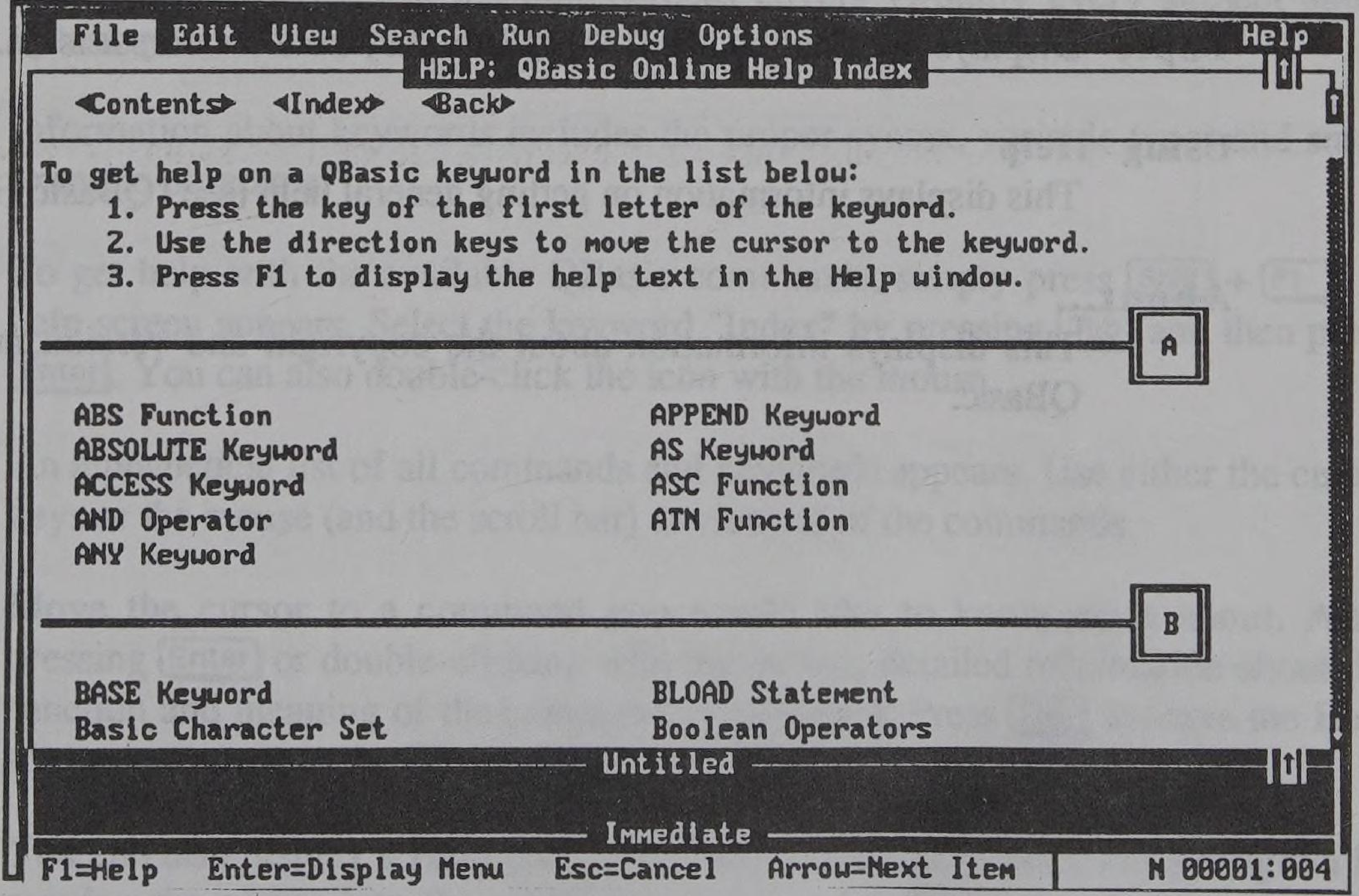
This command is a toggle for turning syntax checking on and off. When activated, QBasic checks each line entered for syntax errors, formats the line and translates the line to executable form if the syntax is correct.


File
Edit
View
Search
Run
Debug
Options
Help
Untitled
Index
Contents
Topic: F1
Using Help Shift+F1
About...
Immediate
F1=Help
Displays help index
N 00001:002

The QBasic Help menu

The Help menu

Index Displays the Help index.


File
Edit
View
Search
Run
Debug
Options
Help
HELP: QBasic Online Help Index
◄Contents► ◄Index► ◄Back►
A
B
Untitled
Immediate
F1=Help
Enter=Display Menu
Esc=Cancel
Arrow=Next Item
N 00001:004

To get help on a QBasic keyword in the list below:

1. Press the key of the first letter of the keyword.
2. Use the direction keys to move the cursor to the keyword.
3. Press F1 to display the help text in the Help window.

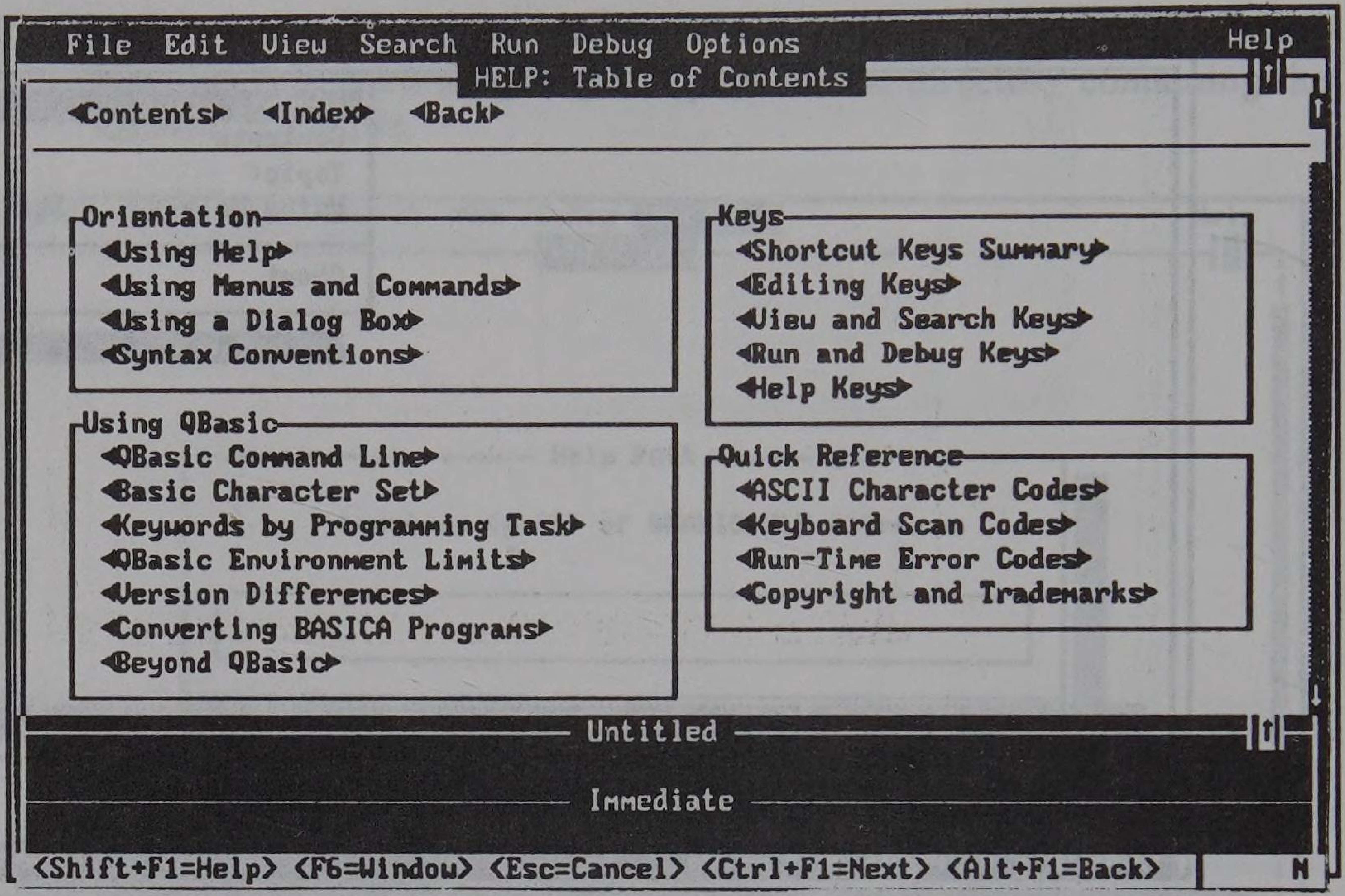
ABS Function
ABSOLUTE Keyword
ACCESS Keyword
AND Operator
ANY Keyword

APPEND Keyword
AS Keyword
ASC Function
ATN Function

BASE Keyword
Basic Character Set

BLOAD Statement
Boolean Operators

The QBasic Help index


File Edit View Search Run Debug Options Help
HELP: Table of Contents
◄Contents► ◄Index► ◄Back►
Untitled
Immediate
<Shift+F1=Help> <F6=Window> <Esc=Cancel> <Ctrl+F1=Next> <Alt+F1=Back>
N

Orientation
◄Using Help►
◄Using Menus and Commands►
◄Using a Dialog Box►
◄Syntax Conventions►

Using QBasic
◄QBasic Command Line►
◄Basic Character Set►
◄Keywords by Programming Task►
◄QBasic Environment Limits►
◄Version Differences►
◄Converting BASICA Programs►
◄Beyond QBasic►

Keys
◄Shortcut Keys Summary►
◄Editing Keys►
◄View and Search Keys►
◄Run and Debug Keys►
◄Help Keys►

Quick Reference
◄ASCII Character Codes►
◄Keyboard Scan Codes►
◄Run-Time Error Codes►
◄Copyright and Trademarks►

Contents of QBasic Help

Contents

Displays **Help** table of contents. This command provides a guide organized according to topics to information in QBasic Help.

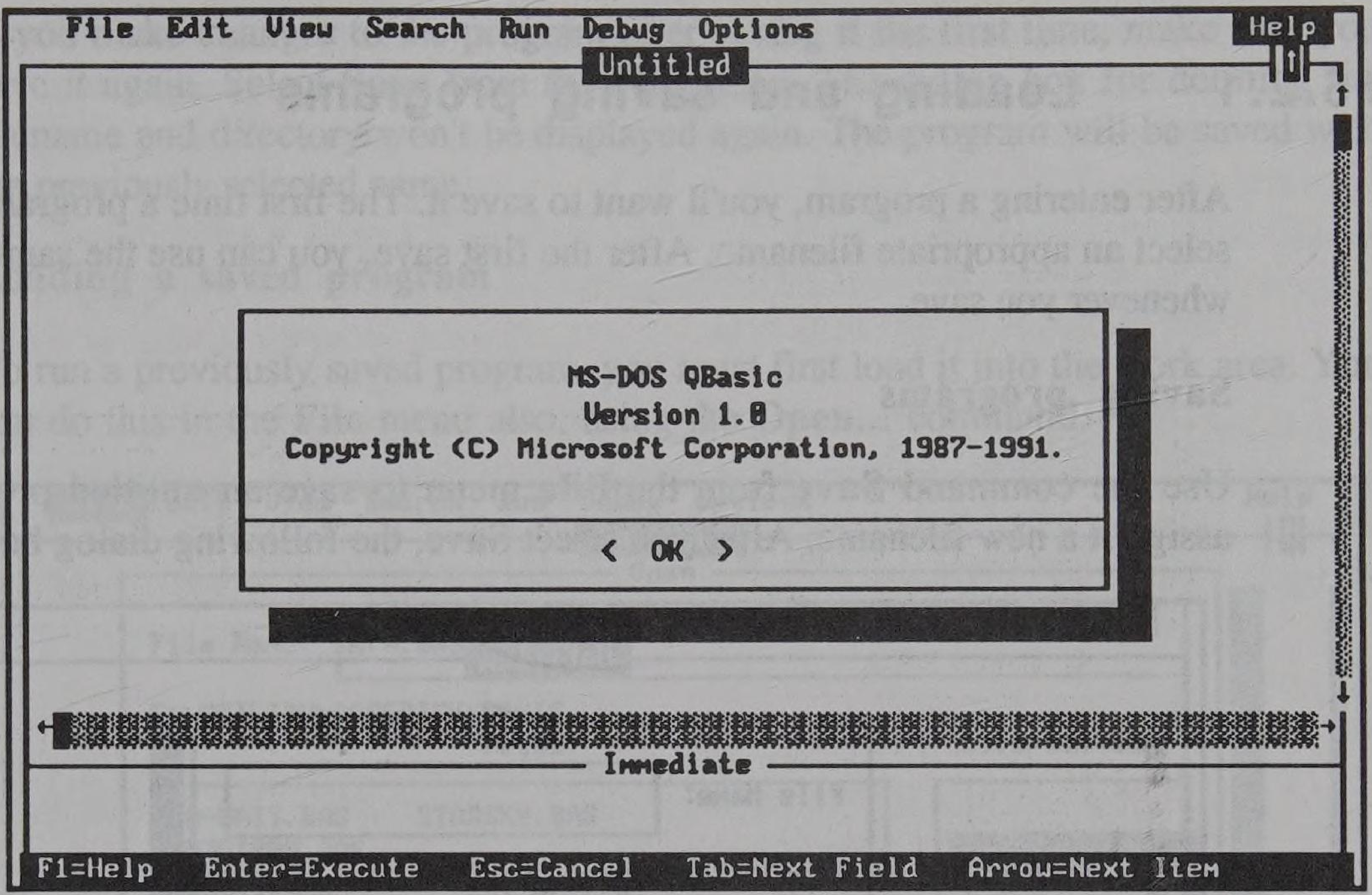
Topic Displays information about the QBasic keyword the cursor is on.

Using Help

This displays information on getting general help (see "QBasic Help").

About...

This displays information about the copyright and version number of QBasic.


File Edit View Search Run Debug Options Help
Untitled
< OK >
Immediate
F1=Help Enter=Execute Esc=Cancel Tab=Next Field Arrow=Next Item

MS-DOS QBasic
Version 1.0
Copyright (C) Microsoft Corporation, 1987-1991.

The About... window

QBasic Help

Using Help may be the most important command of QBasic to understand. The information provided in the help screens covers virtually every subject about QBasic you're likely to encounter.

Information about keywords includes the proper syntax, variable types and small example programs.

To get help with the available QBasic commands, simply press Shift + F1. A help screen appears. Select the keyword "Index" by pressing Tab and then press Enter. You can also double-click the icon with the mouse.

An alphabetical list of all commands and keywords appears. Use either the cursor keys or the mouse (and the scroll bar) to view all of the commands.

Move the cursor to a command you would like to know more about. After pressing Enter or double-clicking with the mouse, detailed information about the function and meaning of the command is displayed. Press Esc to leave the Help screen and return to the QBasic Editor.

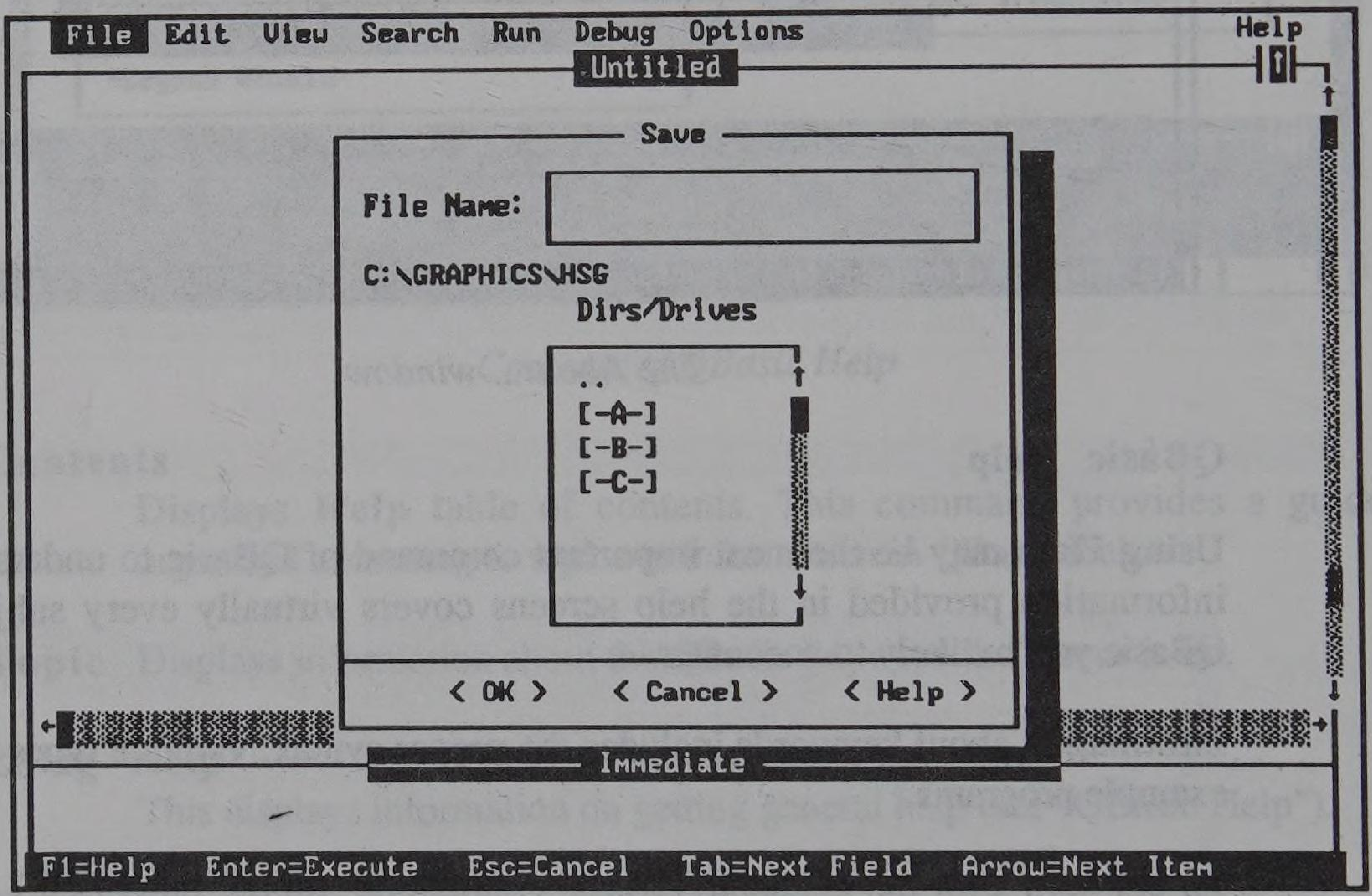
You can also display a Help screen for each command in the current program by moving the cursor into the command and pressing F1. Most help screens for commands even have short sample programs that demonstrate the use of a command.

16.2.1 Loading and saving programs

After entering a program, you'll want to save it. The first time a program is saved, select an appropriate filename. After the first save, you can use the same filename whenever you save.

Saving programs

Use the command **Save** from the **File** menu to save an untitled program and assign it a new filename. After you select **Save**, the following dialog box appears:


File Edit View Search Run Debug Options
Help
Untitled
Save
File Name:
C:\GRAPHICS\HSG
Dirs/Drives
..
[-A-]
[-B-]
[-C-]
< OK >
< Cancel >
< Help >
Immediate
F1=Help
Enter=Execute
Esc=Cancel
Tab=Next Field
Arrow=Next Item

The dialog box for Save

Enter the name under which you want to save the program in the dialog box. After pressing Enter, the program is saved in the current directory of the hard drive or diskette. QBasic automatically attaches .BAS as the filename extension.

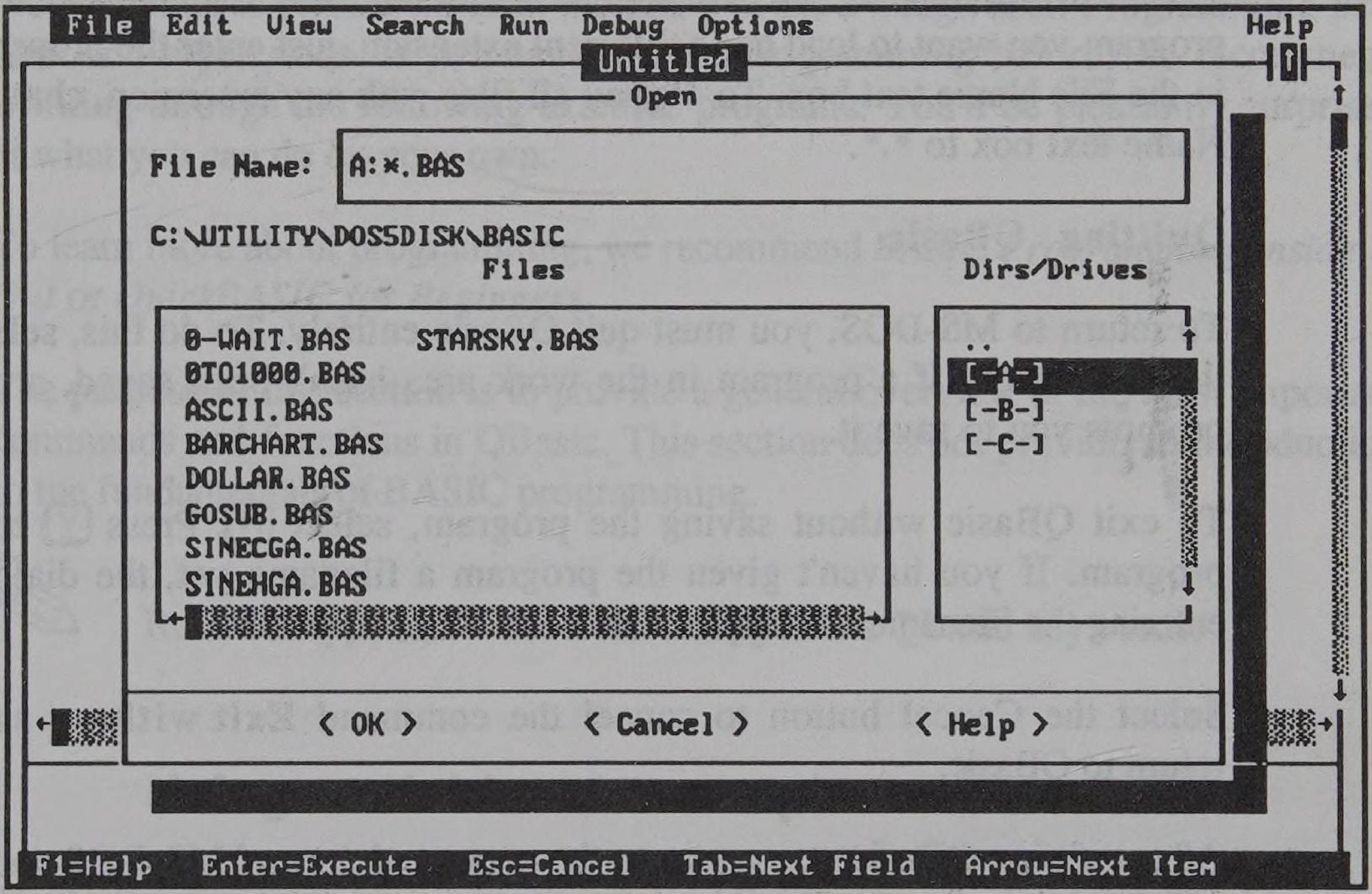
You can also specify the path of the directory where the program is to be saved. Enter the complete path including the filename in the dialog box File Name text box. As an alternative, you can also select the correct path under "Dir/Drives" and then enter just the filename.

When selecting filenames for your programs, remember that filenames can only use eight characters. If a name with more than eight characters is entered, the first three excess characters will be used as the filename extension and all other characters are simply omitted.

If you make changes to the program after saving it the first time, make sure you save it again. Select **Save** from the **File** menu. The dialog box for defining the filename and directory won't be displayed again. The program will be saved with the previously selected name.

Loading a saved program

To run a previously saved program, you must first load it into the work area. You can do this in the **File** menu also, using the **Open...** command.


File
Edit
View
Search
Run
Debug
Options
Help
Untitled
Open
Files
Dirs/Drives
< OK >
< Cancel >
< Help >
F1=Help
Enter=Execute
Esc=Cancel
Tab=Next Field
Arrow=Next Item

File Name: A:*.BAS

C:\UTILITY\DOS5DISK\BASIC

```
0-WAIT.BAS      STARSKY.BAS
0TO1000.BAS
ASCII.BAS
BARCHART.BAS
DOLLAR.BAS
GOSUB.BAS
SINECGA.BAS
SINEHGA.BAS
```

```
..
[-A-]
[-B-]
[-C-]
```

Selecting Open... from the File menu displays this dialog box

Enter the filename (and path, if necessary) of the program in the top line of the dialog box.

There is another way to select a program to be loaded. Enter a wildcard in the File Name text box. This will display all QBasic files in the current directory which satisfy the conditions of the wildcard in the list box.

Press Tab (or use the mouse) to move to the list box. Use the cursor keys (or the mouse) to select the desired file. The filename is automatically placed in the File Name list box.

The drive and path can be changed by using Tab (or the mouse) to select the Dirs/Drives list box and select the desired drive and directory.

You can move to the next highest directory by moving the cursor to the two periods ".." and pressing Enter. To go to the next lowest directory, move the cursor to the directory name (which is always in capitals) and press Enter.

These directories also display the QBasic files. Select the desired file from the list box. After pressing Enter or clicking with the left mouse button, the file is loaded and displayed on the work area.

You can use this method to search your hard drive or diskette for programs, since QBasic always displays all available files with the .BAS filename extension. If the program you want to load has a different extension, just enter the proper extension in the File Name text box. To display all files with any extension, change the File Name text box to *.*.

Quitting QBasic

To return to MS-DOS, you must quit QBasic entirely. To do this, select **Exit** in the **File** menu. If a program in the work area hasn't been saved, a dialog box prompts you to save it.

To exit QBasic without saving the program, select N. Press Y to save the program. If you haven't given the program a filename yet, the dialog box for entering the filename also appears.

Select the Cancel button to cancel the command **Exit without saving** and return to QBasic.

After quitting QBasic you return to the command area of MS-DOS and can enter MS-DOS commands. If you started QBasic from the MS-DOS Shell, you will return to the MS-DOS Shell screen.

16.3 Commands And Functions

Even though QBasic is powerful and easy to use, some programming knowledge is required. A thorough explanation of all of the facets of BASIC programming is beyond the scope of this book. In fact, to adequately cover a subject like QBasic with programming examples and detailed keyword explanations would require a book at least as large as this one.

At first glance it appears complex to learn to program in BASIC, but BASIC programming is certainly within the grasp of the average user. Programming isn't dark and mysterious. If you haven't tried programming before, we recommend working through the following example programs. You'll be pleasantly surprised at what you can do on your own.

To learn more about programming, we recommend *BASIC Programming Inside & Out* or *QuickBASIC for Beginners.*

The purpose of this section is to provide a general overview of the most important commands and functions in QBasic. This section does not provide an introduction to the fundamentals of BASIC programming.

Refer to Appendix F for a full listing of the QBasic keywords.

16.3.1 Using variables for simple calculations

The basis of ordinary programming consists of finding a solution to a problem and describing the solution with the "command" words of the programming language. You can't just think up any solution, but must use the available commands and options of the programming language. This means, a solution in the programming language BASIC is necessarily different than the solution in Pascal or Assembler.

If you have created a program using QBasic commands, use either **Start** from the **Run** menu or press Shift + F5 to start it. QBasic then switches to another screen where all of the program's messages appear. After the program ends, press any key to return to the QBasic Editor.

One of the advantages of every programming language is the fact that it uses variables. Imagine the problems you would encounter if you had to write a separate program for each number whose square root you wanted to calculate. This would be so tedious as to cancel the advantages of programming. Instead of writing a new program for every instance, you can use a general calculation equation with placeholders (the variables) and assign these variables different values for different calculations.

It's not necessary to write a new program every time you calculate a square root. Instead, create a program for calculating the square root of the variable X and then run this program for X=3, X= 4537, and so forth.

Essentially, there are two different types of variables: Numeric variables and alphanumeric variables. Alphanumeric variables are better known as string variables.

Simply put, numeric variables can only contain numeric values, and you can perform any kind of calculation with them.

Strings can consist of almost any character, but you can't perform calculations with them.

Choosing a plain, clear variable name makes a program more understandable. The variables in QBasic can have names up to 40 characters long. This makes it possible for users to give their variables very meaningful, expressive names. However, there are a few restrictions when choosing variable names:

- Names must be at least one character long and can be up to 40 characters long.
- Names can only contain letters (a-z) and numerals (0-9). Spaces are forbidden just as in filenames, since they serve as separators.
- Names cannot start with a numeral and shouldn't contain any QBasic keywords (such as PRINT).

Alphanumeric variables (character strings) have the character "$" at the end of the variable name. Character strings themselves must be defined in quotation marks, to distinguish them from numbers. 147 is a number, while "147" is a character sequence of three alphanumeric characters. In this case the numbers "1", "4" and "7".

Numeric variables can also contain different kinds of numbers. The exact type of variable is indicated by a special character at the end of the variable name. The integer variables are especially important and can contain only whole numbers from -32768 to 32267 and have a percentage sign following the variable name.

The practical thing about numeric variables is that you can use them just like numbers in all arithmetic equations. The four fundamental arithmetic operations (addition, subtraction, multiplication and division) are easy to use in QBasic:

Program line	Output to screen
`PRINT 4 + 9`	13
`PRINT 5 - 3`	2
`PRINT 3 * 5`	15
`PRINT 7 / 3`	2.333333

Examples of variables

Here are a few examples that should explain how to use variables and variable names correctly:

Strings/Character strings

`Name$ = "Miller"`	Character string with 6 characters
`Character$ = "a"`	Character string with 1 character
`Street$ = "25 Star Street"`	Character string with 14 characters, spaces and numbers
`Number$ = "119"`	Character string with 3 characters/numbers

Numeric variables

`Number = 119`	Single precision number
`New_number = 9.5`	Simple precision number
`Number% = 9.5`	Integral number (whole number) with the content of 9, the remainder is cut off during the allocation

16.3.2 Input and Output of data

Commands provide a way to enter data into a program and control the way a program runs. In most cases, you'll want to display the results of your program on the screen. QBasic also has methods for displaying data as well.

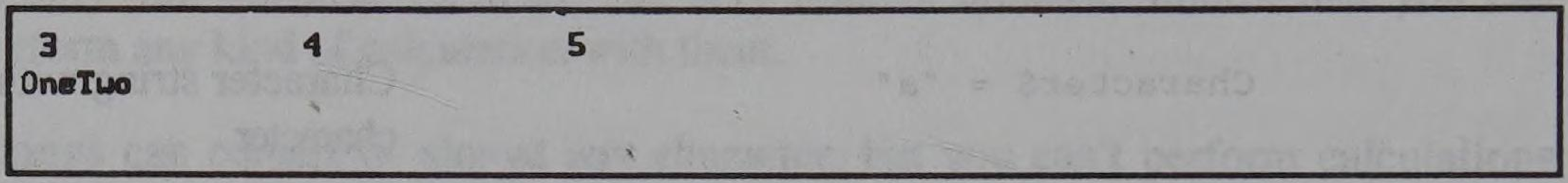
Using PRINT to output data

The PRINT command sends data to the screen. Numbers, character strings or variable names may be specified after PRINT. When a variable name is used, the screen displays the contents of the variable.

```
PRINT 3
PRINT "Three"
Number = 3
PRINT Number
```

QBasic moves the cursor to a new line after each PRINT statement. To prevent this, end the line with either a comma or a semicolon. After a comma, the next information printed to the screen is displayed at the next tab position. When a semicolon is used, the next data printed to the screen follows immediately. In this way, you can output several items with a single PRINT statement:

```
PRINT 3,4,5
PRINT "One";"Two"
```



```
 3             4             5
OneTwo
```

Screen display from the program lines shown above

Notice that in the figure above that the commas in the first line positioned the cursor at the next tab position. Also, the semicolon in the second line printed the two strings without a space between them.

Using INPUT to put data in a variable

The INPUT command can be used to pause a program to get input from the keyboard. The computer displays a question mark to indicate that it is expecting input:

```
INPUT Number
```

The variable type used in the INPUT line determines the type of input that is permitted. In the example above, only numbers can be input. Entering "Miller" would result in an error message and you would have to enter something else. On the other hand, if you specified a character string variable after INPUT, almost any kind of input would be accepted.

Since a question mark prompt doesn't provide information about the kind of input the program expects, you can insert comments, in quotation marks. The variable name is separated from the comments by a semicolon:

```
INPUT "What is your name"; Name$
```

This causes the comment to appear in front of the question mark. If you separate the variable name from the comment by a comma instead of a semicolon, the question mark is not displayed.

```
INPUT "Please enter your name: ", Name$
```

Checking the keyboard

In some cases, INPUT is not flexible enough. For example, you must press Enter after the input. Also, there is no way of checking the input for length or special characters.

You can use the special INKEY$ command as an alternative. Actually, INKEY$ is not a command, but rather a string variable updated by QBasic: It always contains the next character read by the keyboard. If there is no character available, INKEY$ contains an empty string "".

For example, to have a program wait for a key to be pressed before continuing, use the following:

```
PRINT "Please press any key..."
Delay:
KBENTRY$ = INKEY$
IF KBENTRY$ = "" THEN GOTO Delay
```

INKEY$ enables limiting input to a specific length. You can count characters and automatically stop when the maximum limit is reached.

16.3.3 Commands and functions

We've discussed some simple rules for arithmetic. Up until now we have learned arithmetic rules as commands. By using the instruction:

```
Number = 3 + 4
```

QBasic calculates the result and assigns it to the variable "Number". A variable name can also appear on the right side of the equation:

```
Number = Number + 1
```

By now it should be clear that the equals sign isn't really an "equals" sign as expressed in mathematics, but represents an assignment. You are assigning the value on the right of the equals sign to the variable on the left.

QBasic consists of commands and functions. Functions play an important role in programming. You should know the fundamental differences between commands and functions.

Commands are used to have QBasic perform actions, while functions require QBasic to supply a value or result.

For example, to determine the character with an ASCII code of 33, you can use the CHR$() function.

```
PRINT CHR$(33)
```

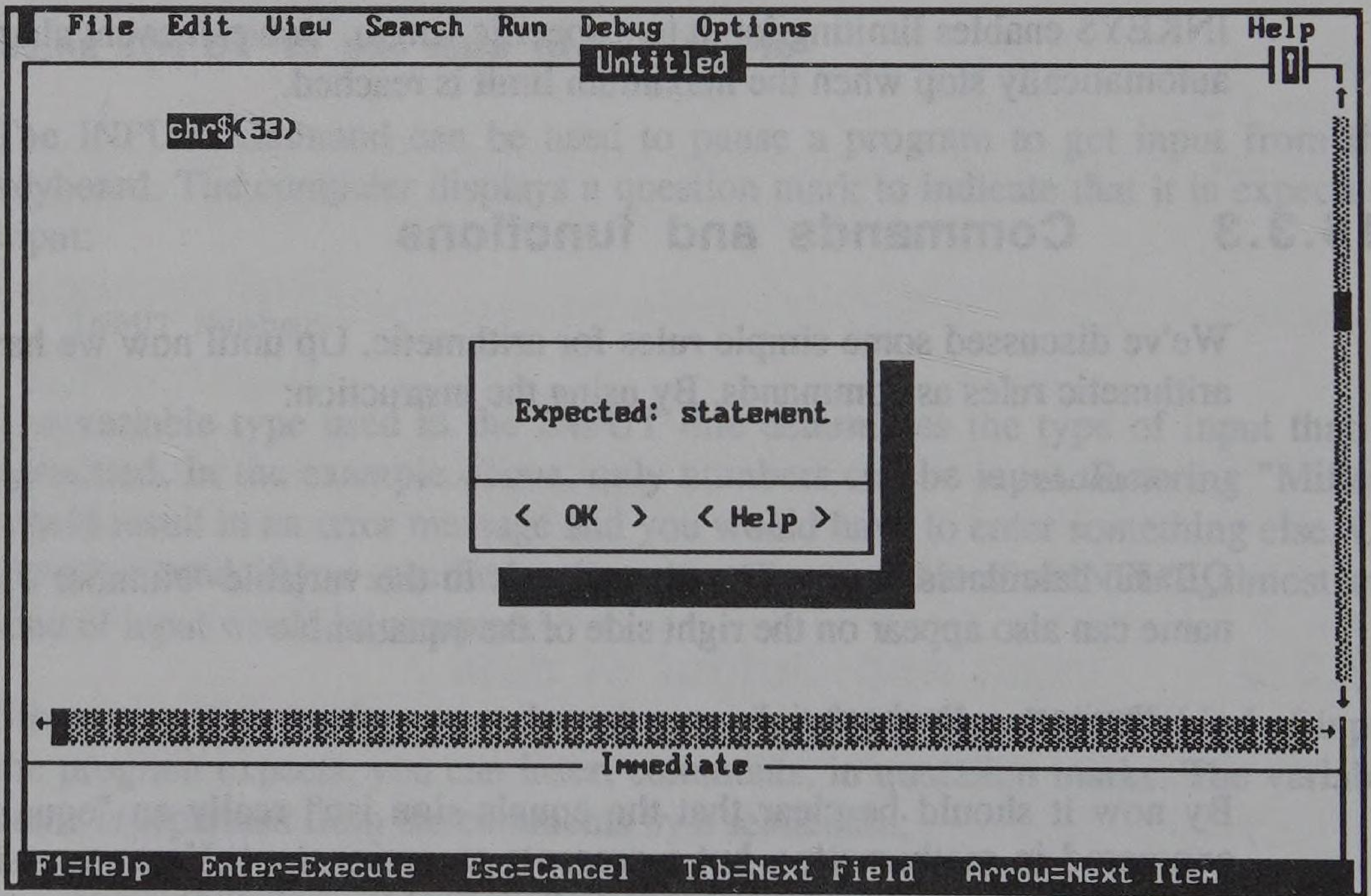
The exclamation point is displayed as a result of this function. More often, functions are used in an assignment and then the result is assigned to another variable:

```
Character$=CHR$(33)
```

Many functions (like CHR$() in our example) require at least one piece of information to be able to work at all. The function CHR$(), for example, requires the ASCII code of the character. These values are specified within parentheses.

When a function like CHR$() is used, there are two things to keep in mind: Usually you must specify a value within the parentheses and you must use the result (function value).

For example, you could display the result on the screen, assign it to a variable and so forth.


File
Edit
View
Search
Run
Debug
Options
Help
Untitled
Immediate
< OK >
< Help >
F1=Help
Enter=Execute
Esc=Cancel
Tab=Next Field
Arrow=Next Item

```
chr$(33)
```

Expected: statement

A function entered in a command line without an assignment results in an error message

16.3.4 Program control

If it weren't for program loops, a program would always run from the first line to the last line. QBasic uses commands that control a program so that some commands can be executed several times.

Conditions: IF ... THEN

You can use the IF command to check for a condition. Commands which are executed if the specified condition is satisfied are written after the THEN portion of the statement.

The following example program converts U.S. dollar amounts into DM (German marks) after the current exchange rate for the dollar has been entered at the beginning of the program. The program continues to execute until a zero is entered as a value for dollar:

```
CLS
INPUT "Please enter rate of exchange: 1 U.S. dollar = ?? DM:
", Rate

Calc:
INPUT "Please enter desired amount in dollars (Enter 0 to
end): "; Dollar
DM = Dollar * Rate
PRINT "$";Dollar; " equals "; DM; " DM"
IF Dollar > 0 THEN GOTO Calc
```

This program checks whether a value greater than zero has been entered. If the value is greater than zero, the rest of the line (after THEN) runs, and the program starts over again at "Calc". (This sample program should be typed in, but for your convenience, it has been placed on the *MS-DOS 5.0 Complete* companion diskette as EXCHANGE.BAS.)

If the condition specified by IF is not fulfilled, QBasic ends the current line and continues in the next line. Since there is no next line, the program ends.

GOTO By the way, the GOTO Calc command means that as soon as QBasic reaches this command, it does not continue line for line, but jumps to the Calc label and continues there. This label must be present, otherwise the program gives you an error message.

Define the label by entering it at the desired place in the program. Labels are a name followed by a colon. More information about GOTO can be found in the section about subroutines.

Conditional options for IF

There are a number of conditional operations used to make comparisons. These operators are called logical operators, because they cause something to happen that demands logical "thinking":

QBasic Logical Operators	
=	equals
<	less than
>	greater than
>=	greater than or equal to
<=	less than or equal to
<>	not equal to

Setting limits: FOR ... NEXT

Many times users want to run a part of a program several times with different numerical values each time. Sometimes it's necessary to run a portion of a program a specified number of times. FOR...NEXT loops are ideal for both of these purposes.

What would be an easy way to display all characters with ASCII codes between 32 and 255 on the screen? Here's a simple solution:

```
CLS
FOR AsciiNumber = 32 TO 255
PRINT CHR$(AsciiNumber);
NEXT AsciiNumber
```

In QBasic, the line "FOR AsciiNumber = 32 TO 255" means that the initial value for AsciiNumber is 32. All of the lines which follow the FOR statement are executed until the line NEXT AsciiNumber is encountered.

The value of AsciiNumber is then increased by "1". If the value of AsciiNumber is not greater than 255, the program "jumps" back to the line containing FOR. (The lines between FOR and NEXT are called a "loop").

This loop begins with 32 because some ASCII characters below this value are not printable characters. This would spoil the results. For example, CHR$(7) emits a warning beep, not a printable character.

Other options

Besides these two options, QBasic has a number of other commands for special uses, such as those in the following list. Look them up as you need them in the appropriate QBasic Help screen.

To get help for a specific command, just type the command word and press F1.

DO... LOOP	Execute as long as or until a condition is fulfilled.
WHILE...WEND	Execute as long as a condition is fulfilled.

SELECT CASE	Run one of several different parts of a program based on the selection of a value.

16.3.5 Using subroutines

Users once complained that BASIC couldn't be used to create structured, easy to read programs. With QBasic, this is no longer the case, partly because of its very efficient, powerful subroutines.

Jumping with GOTO and GOSUB

QBasic ordinarily runs the lines of a program one after the other from beginning to end. You can use the GOTO command to have QBasic continue in a different place in the program. Just mark the target area with a label and GOTO the label.

It should be noted that using GOTO is frowned upon by many programmers. Using GOTO often leads to very inelegant program structures. Overuse of GOTO can lead to programs which are extremely difficult to follow. Using subroutines is the preferred method of program flow control.

One problem with GOTO is that when QBasic jumps to another program line, it no longer knows where it came from.

Quite often it would be convenient to return to the line where you made the jump from. One reason could be that you want to use one part of a program several times. For example, you might want to use the lines for user input or for displaying information on the screen.

The following short program shows you why you can't do that with the GOTO command. The program assigns different values to a variable at the beginning, and then jumps to a small output program, which represents the program section to be repeated:

```
VARIABLE = 10
GOTO Type
Back:
VARIABLE = 20
GOTO Type
VARIABLE = 30
GOTO Type
END
Type:
 PRINT "The variable is "; VARIABLE
 GOTO Back
```

By the way, instead of letting the program run to the last line, the END command in the fourth line from the bottom immediately ends the program.

The program functions correctly the first time. It outputs 10 and sets VARIABLE to 20. After displaying 20, the program jumps back again and outputs 20 a second time. The problem is that VARIABLE can't be set to all of the values, because the program always jumps back to the same line.

What's needed is a way to jump from anywhere in the program to a section that you need often and then return to the original line. You can solve this problem using subroutines together with GOSUB and RETURN.

A subroutine is a part of a program that may be executed numerous times, may be called from anywhere in the program and always returns to the line immediately following the calling line.

Instead of using GOTO, you use GOSUB to jump to the first line of this program section. QBasic remembers it's present position in the file, continues execution at the label, reaches RETURN and goes back to the line immediately following the GOSUB statement. Here's an example:

```
CLS
MARK = 100
GOSUB Calculate
MARK = 500
GOSUB Calculate
MARK = 10
GOSUB Calculate
END
Calculate:
 CLS
 PRINT "                Conversion: DM - Dollars "
 PRINT
 DOLLARS = MARK / 1.5
 PRINT MARK ; " DM are "; DOLLARS ; " Dollars"
 PRINT
 INPUT "Continue (ENTER) "; KBENTRY$
RETURN
```

The subroutine starts in the line with the label CALCULATE and ends in the line with RETURN. It's executed three times. The first time it is called in the third line. MARK was set to 100. The subroutine calculates the appropriate number of DOLLARS and displays it on the screen. After RETURN, QBasic continues in the fourth line. The same thing happens with 500 and 10 in the following lines.

The END at the bottom of the main program, before the subroutine, is important. If the program didn't end here, it would run the subroutine again beginning at the label and would encounter RETURN at the end. Since there is no GOSUB calling the subroutine, QBasic would return an error message.

Subroutines and their practical uses

Although in many cases larger programs can be structured in such a way that they run continuously from "top" to "bottom" (i.e., from the first to the last line), there are several reasons why you wouldn't want to do this. For example:

- readability
- simple variability and expandability
- it's difficult to search for errors

Instead, build programs from program parts and give them a hierarchical structure. It's best if you use small, closed task areas to separate program parts.

Programs that aren't too large begin with a main program and execute different tasks and activities in subroutines. Programs that are very large often have subroutines that use subroutines of their own.

```
REM Start the main program
..
GOSUB SR1
GOSUB SR2
IF Result < 100 THEN GOSUB SR3
GOSUB SR1
END

REM Subroutine 1
SR1:
..
RETURN

REM Subroutine 2
SR2:
..
RETURN

REM Subroutine 3
SR3:
..
RETURN
```

The lines with the periods represent any command lines within the program.

This outline of the program structure clearly shows that the actual work is delegated to the various subroutines, which are called from the main program in the correct sequence.

The outline should also demonstrate that you can very easily make certain tasks dependent on conditions (indicated in the example by "IF Result < 100 THEN GOSUB SR3") and that you can also call a subroutine more than once (SR1 in the example).

The following example illustrates the use of subroutines. In the main program, a counter is increased in the loop and calls the subroutine Delay after displaying to the screen.

This subroutine uses delay loops to delay the program (QBasic counts from 0 to 1000). At the same time, you can terminate the delay loop prematurely by pressing a key.

```
REM Main program
CLS
FOR WaitLoop = 1 TO 10
PRINT WaitLoop
GOSUB Delay
NEXT WaitLoop
END

REM Subroutine Delay
Delay:
Counter = 0
WHILE INKEY$ = "" AND Counter < 1000
Counter = Counter + 1
WEND
RETURN
```

More options using SUB and FUNCTION

QBasic has other structures that are even more powerful. These structures start with SUB, end with END SUB and are called "procedures". You can start procedures like commands, by entering the name of the procedure. This allows you to expand QBasic by adding your own commands. Here's what our sample program from above would look like with the Delay procedure:

```
DECLARE SUB Delay ()
CLS
FOR Y = 1 TO 10
PRINT Y
Delay
NEXT Y
END

SUB Delay
```

```
Delay:
Counter = 0
WHILE INKEY$ = "" AND Counter < 1000
Counter = Counter + 1
WEND
END SUB
```

At the beginning of the program you have to tell QBasic that the program contains a Delay procedure. After that you can call it in the main program. The procedure follows at the end of the main program.

QBasic usually divides the program into the main program and its procedures, displaying only one coherent part of the program.

To enter the program above, you ordinarily use the **New SUB...** command from the **Edit** menu in the (empty) BASIC editor. QBasic will prompt you for the name of the procedure. Enter Delay. The beginning and end lines of a procedure appear on the screen:

```
SUB Delay
END SUB
```

In between these two lines, enter the lines for the procedure from the example above. To enter the actual main program, select **View SUBs...**. This displays the existing elements of the program. The first line in the display is the name of the main program, Untitled. Select this command and press Enter to display it on the screen. Then enter the name of the main program.

Just as you can use **New SUB...** to create new subprograms, you can use **New FUNCTION...** from the **Edit** menu to create new functions in QBasic. A typical example of such a new function could be a subroutine that waits for the press of a key and then delivers this keypress to the main program.

16.4 QBasic And Graphics

QBasic has some powerful options for creating graphics. The sample programs that are supplied with MS-DOS 5.0 demonstrate that you can even use these options to create computer games.

Graphics made of ASCII characters

To be able to take advantage of the QBasic graphics commands, you need a graphics card which is a special expansion to your PC that corresponds to the IBM standard. However, different graphics cards vary considerably in the number of dots that can be displayed (resolution).

If you want to program QBasic programs with special graphics commands, the programs frequently have to do complicated calculations so that the results on different computers are identical.

There is another option for creating graphics that is a lot easier because it doesn't use either a graphics card or the special QBasic graphics commands. This option uses ASCII characters. ASCII is the acronym for American Standard Code for Information Interchange.

A PC can only "think" in numbers and has to translate each character transmitted from the keyboard into one of these numbers. ASCII codes translate into characters. The numbers begin at 49 (Alt + 4 9 =1) and the letters begin at 65 (Alt + 6 5 = A).

The following sample program creates a random bar chart out of the "*" character:

```
CLS
RANDOMIZE TIMER
FOR I = 5 TO 60 STEP 5
 HEIGHT% = 24 - ((22 * RND) + 1 )
  FOR Y = 24 TO HEIGHT% STEP -1
  LOCATE Y,I
  PRINT "****";
 NEXT Y
NEXT I
```

The program looks somewhat complicated because the chart is created randomly and starts at the bottom, working its way to the top. Since you can easily get help and information about unknown commands by placing the cursor on the command and pressing F1, you can find out everything you need to know about the program by using the QBasic online help screens.

It's especially easy to display a trigonometric function with ASCII graphics. The following example displays the sine function on the screen:

```
CLS
FOR I = 1 TO 80
Y = I * 3.14 * 2 / 80
Y = INT(SIN(Y) * 12) + 13
LOCATE Y, I
PRINT CHR$(249);
NEXT I
WHILE INKEY$ = "": WEND
```

The most difficult thing about this program is the "scaling" (i.e., the conversion to the screen coordinates). Since the sine repeats itself after 2 PI, this range of numbers is divided into the 80 columns of the screen.

16.4.1 Different graphics modes

In our last example we confined ourselves to ASCII graphics, since anyone can use this form of graphics regardless of the graphics capacities of their PC. On the other hand, if you do have a PC with a graphics card, QBasic provides some powerful commands for high resolution graphics. For example, one interesting option involves changing the screen display from 640 pixels in width and 200 pixels vertically.

Since lines and circles are made up of many pixels and are a lot of trouble to make from individual pixels, QBasic provides special commands for making them. For example, with graphics commands it's easy to display mathematical functions on the screen in graphics.

There are two things to keep in mind when using these graphics options:

1. Depending on the kind of graphics card, there can be a considerable difference in the commands for turning on graphics and the maximum number of pixels (resolution) in the screen display.

2. If your PC has a Hercules Graphics card, start the MSHERC.COM program before starting QBasic. Displaying graphics is impossible otherwise.

16.4.2 Basic graphics commands

The easiest way to learn the basic graphics commands and the correct sequence is through examples. In the previous section we showed you a sine in text mode (ASCII graphics).

For comparison, here is the same sine function using the graphics options of QBasic. We added a “V” in front of the variable names since WIDTH is a QBasic keyword:

Example of the sine in Hercules Graphics

```
VMode = 3
VHeight = 348
VWidth = 720

SCREEN VMode
CLS
FOR I = 1 TO VWidth
Y = I * 3.14 * 2 / VWidth
Y = INT(SIN(Y) * VHeight / 2 + VHeight / 2)
PSET (I, Y), 15
NEXT I
WHILE INKEY$ = "": WEND
```

Example of the sine function with EGA/VGA Graphics

For EGA/VGA cards with 640 * 350 pixels, change the following lines:

```
VMode = 9
VHeight = 350
VWidth = 640
```

You must use the SCREEN command to switch the screen to graphics mode. The kind of mode that is possible depends a great deal on the kind of graphics card and results in very different pixel display widths and heights.

The PSET command actually draws the pixels.

QBasic can do a lot more than just draw pixels. You can use LINE to create lines and rectangles and you can draw circles with the CIRCLE command.

16.4.3 Preserving your screen

In conclusion, here's a screen saver program. You should not allow the screen to display a motionless image over a long period of time. This can leave "traces" in the phosphorus layer of the screen. This is called "burning in" an image.

Although you can create a screen saver in two simple BASIC lines:

```
CLS
WHILE INKEY$="":WEND
```

a screen that is completely empty can be rather boring.

The following program (on the companion diskette under the name STARSKY.BAS) draws a "colorful starry sky" until you press a key.

```
REM Starry sky as a screen saver
DEFINT A-Z
RANDOMIZE TIMER
VidMode = 9
VHeight = 350
VWidth = 640

SCREEN VidMode
WHILE INKEY$ = ""
   Color = RND * 15
   y = VHeight * RND
   x = VWidth * RND
   PSET (x, y), VColor
WEND
SCREEN 0
```

If you have a Hercules Graphics card and want to use this program, change the following lines:

```
VidMode = 3
VHeight = 348
VWidth = 720
```

Remember that you have to start the MSHERC.COM program before starting QBasic if you have a Hercules Graphics card.

To use the screen saver, start QBasic, load and start the program. Create a small batch file called PRESERVE.BAT, containing the following line:

```
QBasic /RUN C:\BASIC\STARSKY.BAS
```

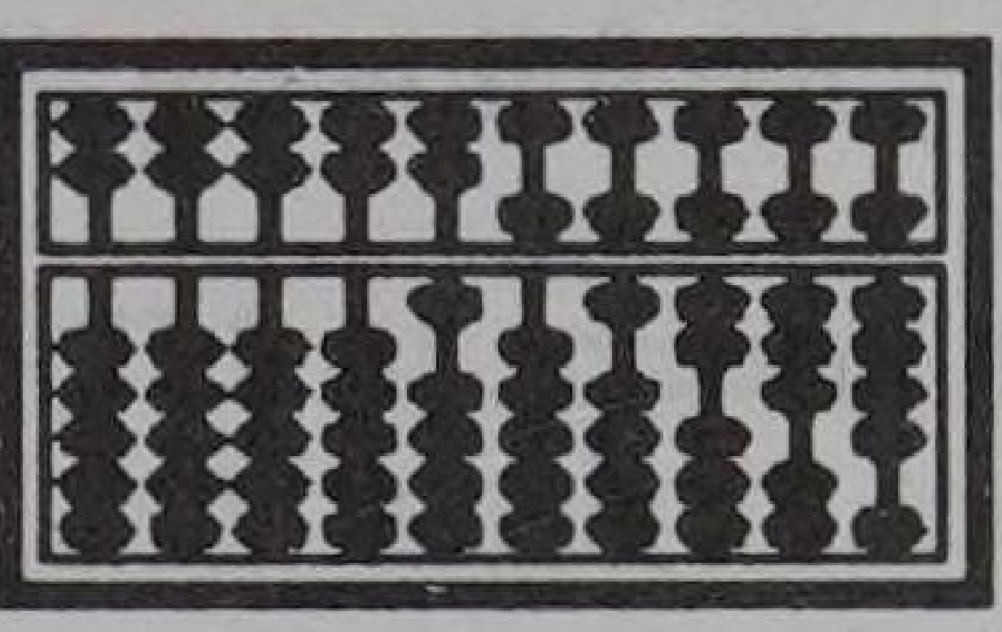
The BASIC program must be in the specified directory, otherwise you will need to change the line.

If you want to exit QBasic automatically at the end of the program so that you can continue working in the command interpreter, just add the following line after SCREEN 0:

```
SYSTEM
```

If you had fun with BASIC programming, you'll find a few additional examples in Chapter 17 that are easy to do in QBasic.

Chapter



17

MS-DOS Tips And Tricks

This chapter contains a collection of practical and useful techniques for working with MS-DOS 5.0. Although we've named the chapter MS-DOS Tips And Tricks, once you understand them, there isn't anything tricky about the techniques. But we're sure that they'll be helpful in making your computing sessions more productive.

This book includes a companion diskette. All the files described in this chapter can be found in the TIPTRICK directory of this companion diskette. Some of these files access EXE and COM programs found in the TOOLS directory.

The TINSTALL program installs all files on the companion diskette, and adapts AUTOEXEC.BAT and CONFIG.SYS to access these files. Please see the Appendices for more information on installing the companion diskette files.

Important notes on how the following tips and batch files function:

1. The device driver ANSI.SYS must be loaded from CONFIG.SYS, specifying the correct path. For example, if ANSI.SYS is in the C:\DOS directory of the hard drive, the following line should be included in your CONFIG.SYS file:

```
DEVICE=C:\DOS\ANSI.SYS
```

2. The SHELL command in CONFIG.SYS must load COMMAND.COM into memory using a larger than usual environment allocation. For example, if COMMAND.COM is in the C:\DOS directory, a line similar to the following is needed in your CONFIG.SYS file:

```
SHELL=C:\DOS\COMMAND.COM /P /E:1000
```

3. The TINSTALL program on the companion diskette installs all files and adds initialization data to direct your system to the subdirectories it creates. The files for this chapter are all located in a directory named \TIPTRICK. The PATH command should include this directory. This lets you start the files from any directory.

4. You should consider adding the option of cancelling file or diskette access by including the following line in your AUTOEXEC.BAT file:

```
BREAK ON
```

5. Hints for typing batch files by hand:

☞ Individual command lines as printed in this book must be typed on a single line. Some lines will wrap around to the next line of text because of the width of the printed page, but must still be entered as a single line.

☞ ECHO. (including a period) in a batch file sends a blank line to the output device.

6. Short batch files can be entered using COPY CON. You can enter new batch files and edit existing ones using EDLIN, the MS-DOS Editor or a word processor that can write text as an ASCII file. Some batch files and data files are more easily entered with one method than another (e.g., COPY CON or EDLIN may be preferable to using the MS-DOS Editor).

7. You can enter the decorative frame characters within a batch file by pressing the following:

```
Alt+201 ╔ = Alt+205  ╦  Alt+203  ╗ Alt+187
Alt+186 ║                        ║ Alt+186
Alt+204 ╠                ╬ Alt+206 ╣ Alt+185
Alt+186 ║                        ║ Alt+186
Alt+200 ╚ = Alt+205  ╩  Alt+202  ╝ Alt+188
```

17.1 An Instant ASCII Table

Have you ever wanted to enter a special character using Alt and the numeric keypad, but couldn't remember the ASCII code for it? Try this.

We created an ASCII table and used a batch file to copy it onto the screen. We could enter the table by hand using a word processor, but it's a lot easier to let a BASIC program generate the table for us.

The following QBasic program, ASCII.BAS, creates a table named ASCII.TXT in the current directory. The ASCII.TXT file is in the C:\BAT directory on the companion diskette:

ASCII.BAS

```
REM ASCII.BAS: Create ASCII table!
OPEN "ASCII.TXT" FOR OUTPUT AS #1
FOR i = 32 TO 255 STEP 12
  FOR y = 0 TO 11
    IF i + y < 99 THEN PRINT #1, " ";
    IF i + y < 256 THEN PRINT #1, i + y; CHR$(i + y);
  NEXT y: PRINT #1,
NEXT i
CLOSE #1
END
```

Enter this program in QBasic and run it. It creates the ASCII.TXT file.

Now we need a short batch program to copy the ASCII table onto the screen. Here's what the ASCII.BAT program looks like:

ASCII.BAT

```
@REM ASCII.BAT
@REM ╔═══════════════════════════════════════════════════╗
@REM ║ Output ASCII table onto the Screen                ║
@REM ║ NOTE: ASCII.TXT must be in same dir as this file  ║
@REM ╚═══════════════════════════════════════════════════╝
@ECHO off
CLS
TYPE ASCII.TXT
@ECHO on
```

Now, when you want to find the value for a character, just type:

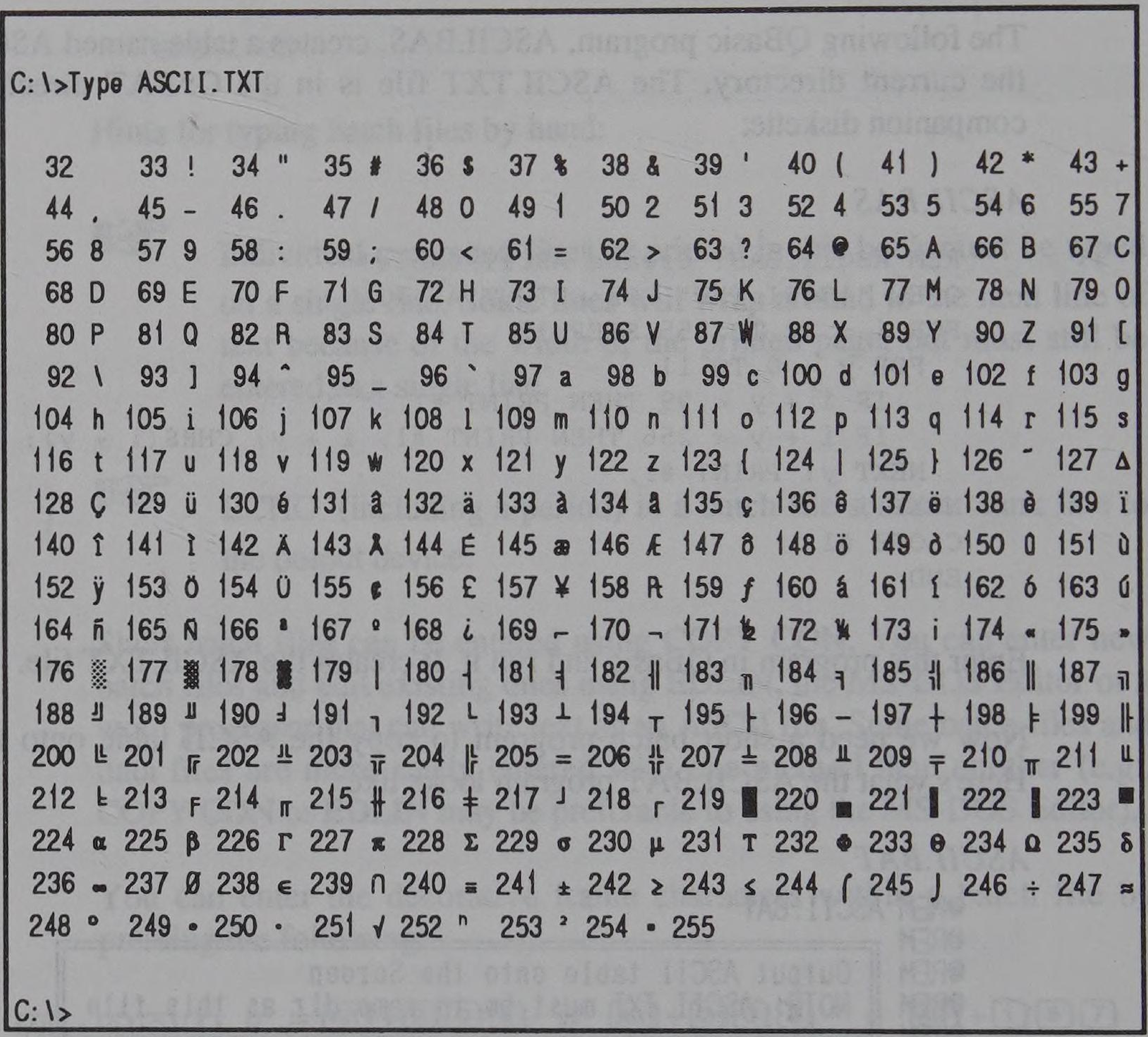
```
ASCII
```

Using the CLS command before typing the file on the screen speeds it up. You may have to add the full path in the TYPE command, depending on where you store the ASCII.TXT file.

Entering the QBasic program takes a lot less time than looking up the table in a manual, and once the QBasic program is entered and run, the information is immediately available.

Remember to use ECHO OFF to turn off the additional command line output.

Here's the file ASCII.TXT, generated by the ASCII.BAS program, as it appears when you use the TYPE command to display it:



```
C:\>Type ASCII.TXT

 32      33 !  34 "   35 #   36 $   37 %   38 &   39 '   40 (   41 )   42 *   43 +
 44 ,   45 -   46 .   47 /   48 0   49 1   50 2   51 3   52 4   53 5   54 6   55 7
 56 8   57 9   58 :   59 ;   60 <   61 =   62 >   63 ?   64 @   65 A   66 B   67 C
 68 D   69 E   70 F   71 G   72 H   73 I   74 J   75 K   76 L   77 M   78 N   79 O
 80 P   81 Q   82 R   83 S   84 T   85 U   86 V   87 W   88 X   89 Y   90 Z   91 [
 92 \   93 ]   94 ^   95 _   96 `   97 a   98 b   99 c  100 d  101 e  102 f  103 g
104 h  105 i  106 j  107 k  108 l  109 m  110 n  111 o  112 p  113 q  114 r  115 s
116 t  117 u  118 v  119 w  120 x  121 y  122 z  123 {  124 |  125 }  126 ~  127 Δ
128 Ç  129 ü  130 é  131 â  132 ä  133 à  134 å  135 ç  136 ê  137 ë  138 è  139 ï
140 î  141 ì  142 Ä  143 Å  144 É  145 æ  146 Æ  147 ô  148 ö  149 ò  150 û  151 ù
152 ÿ  153 Ö  154 Ü  155 ¢  156 £  157 ¥  158 ₧  159 ƒ  160 á  161 í  162 ó  163 ú
164 ñ  165 Ñ  166 ª  167 º  168 ¿  169 ⌐  170 ¬  171 ½  172 ¼  173 ¡  174 «  175 »
176 ░  177 ▒  178 ▓  179 │  180 ┤  181 ╡  182 ╢  183 ╖  184 ╕  185 ╣  186 ║  187 ╗
188 ╝  189 ╜  190 ╛  191 ┐  192 └  193 ┴  194 ┬  195 ├  196 ─  197 ┼  198 ╞  199 ╟
200 ╚  201 ╔  202 ╩  203 ╦  204 ╠  205 ═  206 ╬  207 ╧  208 ╨  209 ╤  210 ╥  211 ╙
212 ╘  213 ╒  214 ╓  215 ╫  216 ╪  217 ┘  218 ┌  219 █  220 ▄  221 ▌  222 ▐  223 ▀
224 α  225 β  226 Γ  227 π  228 Σ  229 σ  230 µ  231 τ  232 Φ  233 Θ  234 Ω  235 δ
236 ∞  237 Ø  238 ∈  239 ∩  240 ≡  241 ±  242 ≥  243 ≤  244 ⌠  245 ⌡  246 ÷  247 ≈
248 °  249 •  250 ·  251 √  252 ⁿ  253 ²  254 ■  255

C:\>
```

ASCII.TXT output

17.2 Searching For Files

Have you ever searched the entire hard drive for a file because you had forgotten in which directory you'd stored it?

The WHERFILE.BAT batch file searches the current drive, including all of the subdirectories, for a specified filename, directory name or part of a name.

WHERFILE.BAT

```
@REM WHERFILE.BAT
@REM ╔═══════════════════════════════════╗
@REM ║ Searches current drive for a file ║
@REM ╚═══════════════════════════════════╝
@ECHO off
IF "%1" == "" GOTO Syntax
DIR \%1 /S /P
GOTO EndItAll
:Syntax
ECHO Searches current drive for a file
ECHO Call - Filename (with wildcards)
:EndItAll
ECHO ON
```

To search for a file named BOOK.TXT, enter WHERFILE BOOK.TXT. To track all files beginning with BO, enter WHERFILE BO*.*. Entering WHERFILE .TXT tells the batch file to search for all files with a .TXT extension. Notice the space between WHERFILE and extension of the files to be searched for.

Examples of calling WHERFILE.BAT

WHERFILE .TMP	Finds all temporary files.
WHERFILE .BAS	Locates all files with the .BAS filename extension.

17.3 Bypassing Error Messages

In addition to the /C, /P and /E switches used for calling the command interpreter, there is also a "secret" switch (/F) which can be very useful in certain situations.

If you use this switch to call COMMAND.COM, the error message and prompt, "Abort, Retry, Fail" still appears when an error occurs, but you no longer have to press a key. COMMAND.COM acts as though the F key (for "Fail") had been pressed.

You can take advantage of this option when you want to copy all of the files from a defective diskette, but don't want to press a key after each error message.

RELEASF.BAT calls a command interpreter with the /F switch for an MS-DOS command and then switches back to the original COMMAND.COM.

RELEASF.BAT

```
@REM ╔══════════════════════════════════════════╗
@REM ║ RELEASF.BAT: COMMAND.COM without waiting ║
@REM ╚══════════════════════════════════════════╝
@ECHO off
COMMAND /f /c %1 %2 %3 %4 %5
@ECHO on
```

To use this batch file, you have to enter the desired command after the name of the batch file when calling it (e.g., RELEASF DIR B: to read the directory of a disk in drive B: that may be slightly corrupted).

When you specify /F, COMMAND.COM confirms errors messages by itself, while specifying /C removes this version of the command interpreter from memory after the commands are finished running.

For example, to copy all of the files from a diskette in drive A: onto drive B:, without having to confirm error messages using the keyboard, type the following command:

```
RELEASF COPY A:*.* B:
```

17.4 Applying Commands

It's too bad that many MS-DOS commands don't let you use wildcards to specify filenames. TYPE and FIND are two such commands. A special command for batch programming can remedy the situation: FOR...DO.

The following batch file, DO.BAT, provides a solution:

DO.BAT

```
@REM ╔═══════════════════════════════════════╗
@REM ║ DO.BAT: Using commands on files       ║
@REM ╚═══════════════════════════════════════╝
@ECHO off
IF "%1" == "" GOTO error
IF "%2" == "" GOTO error
@FOR %%a IN (%2) DO %1 %%a %3 %4 %5
GOTO end:
:error
ECHO Call: command filename (parameter)
:end
@ECHO on
```

The FIND command in MS-DOS requires special handling, which we'll discuss below. First, we want to show you a few standard applications of DO.BAT:

```
DO TYPE *.BAT
```

Displays the contents of all batch files in the current directory on the screen, in directory order.

```
DO DEL *.*
```

Deletes all the files in the current directory without confirmation (DO NOT ENTER THIS COMMAND UNLESS YOU WANT TO DELETE ALL FILES).

DO.BAT doesn't work with the FIND command on its own. This is because FIND expects the search text to precede the filename. If we called our batch for FIND, for example, with:

```
DO FIND SearchText *.BAT
```

then the command would use SearchText as the filename and *.BAT as the parameter. DOF.BAT takes this special requirement of the FIND command into consideration:

DOF.BAT

```
@REM ╔═══════════════════════════════════╗
@REM ║ DOF.BAT: Using FIND on all files  ║
@REM ╚═══════════════════════════════════╝
@ECHO off
IF "%1" == "" GOTO error
IF "%2" == "" GOTO error
@FOR %%a IN (%2) DO FIND "%1" %%a
GOTO end:
:error
ECHO Call: searchtext filename
:end
@ECHO on
```

Let's say you want to search for the CALL command in all of the batch files of the current directory. Just type the following command line:

```
DOF CALL *.BAT
```

The batch file then displays all of the files it searched for along with the lines containing the CALL command.

You can also enter the FOR...DO command outside of a batch file directly after the command interpreter's system prompt. Then you just need to specify one percent sign (%a) rather than two (%%a).

17.5 Screen Header Using Date And Path

Some MS-DOS users like to have as much information on the screen as possible. Others prefer simplicity and don't care to have a long system prompt on the screen. The batch file HEADER.BAT gives you both readability and a lot of information:

HEADER.BAT

```
@REM
@REM  A header with the date and the path  with ANSI.SYS)
@REM
@PROMPT $e[s$e[f$e[7m$t$h$h$h$h$h$h time $d $p$g$e[K$e[u$e[0m$p$g
```

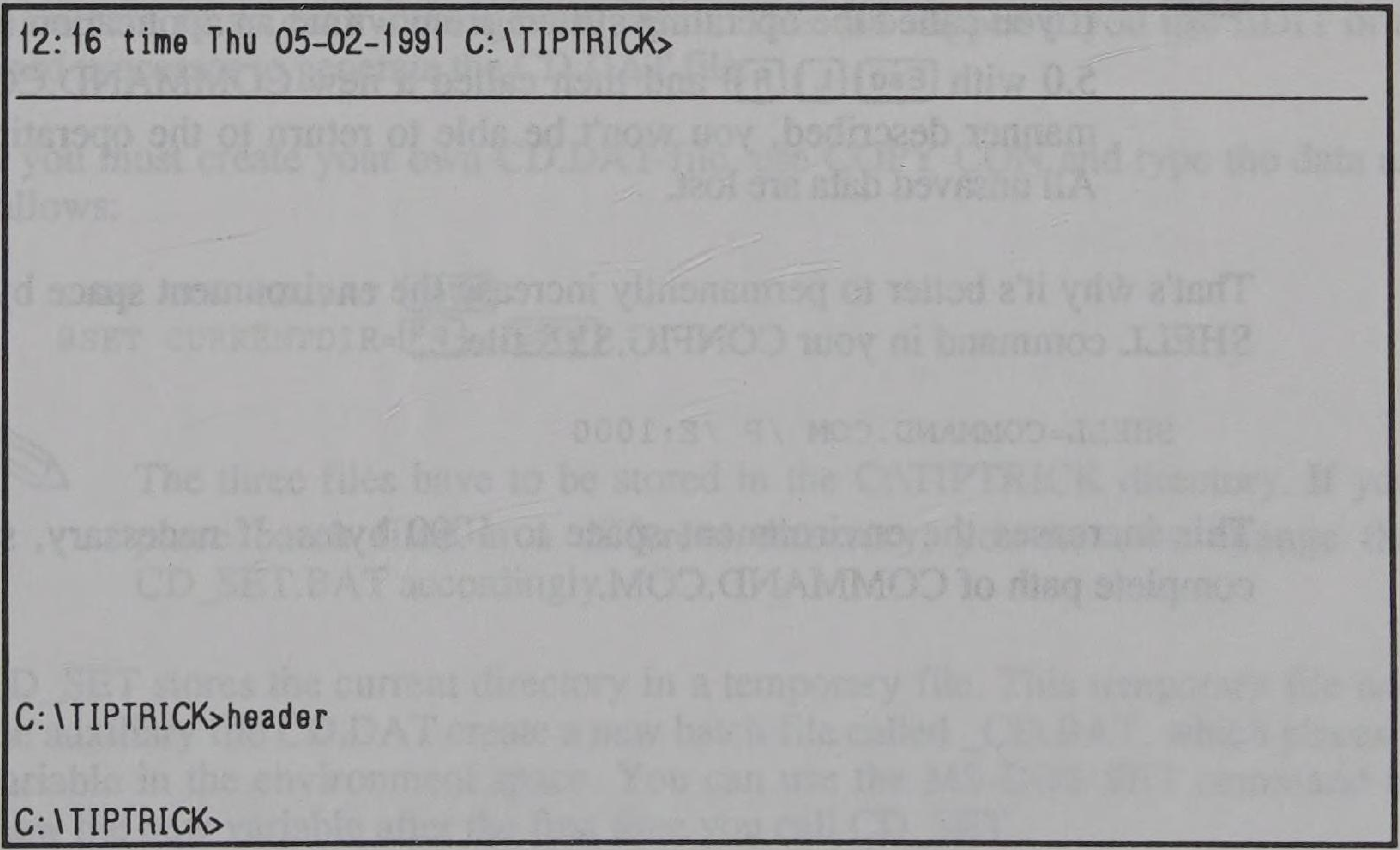
To remove the header, just enter a PROMPT command stating your preferred prompt style, and another command to scroll data or clear the screen. For example:

```
PROMPT $P$G
CLS
```

You must add the screen driver ANSI.SYS to your CONFIG.SYS for this batch file to work properly. Type the following line in CONFIG.SYS:

```
DEVICE=ANSI.SYS
```

If necessary, specify your MS-DOS directory from the hard drive. Or else make sure you have ANSI.SYS on your boot diskette.



```
12:16 time Thu 05-02-1991 C:\TIPTRICK>


C:\TIPTRICK>header

C:\TIPTRICK>
```

HEADER.BAT output

17.6 Increasing Environment Space

The MS-DOS environment can be altered by using system variables. The most important system variables are:

PATH	Contains the entire path specification.
PROMPT	Contains the system prompt.
COMSPEC	Contains the command interpreter path and name.

If you decide to extend a system variable (e.g., lengthen the path), the amount of memory for this reserved by MS-DOS may no longer be sufficient. Then the following error message appears:

```
Out of environment space
```

Increasing the environment space

To create a larger system environment, you can call a new COMMAND.COM with the /P /E:1000 switches. /P specifies that the command interpreter should be loaded permanently and completely replace the original one, /E sets the environment space's size to the specified value in bytes.

If you called the operating system from within an application (e.g., Word 5.0 with Esc L R) and then called a new COMMAND.COM in the manner described, you won't be able to return to the operating system. All unsaved data are lost.

That's why it's better to permanently increase the environment space by using the SHELL command in your CONFIG.SYS file:

```
SHELL=COMMAND.COM /P /E:1000
```

This increases the environment space to 1000 bytes. If necessary, specify the complete path of COMMAND.COM.

17.7 Returning To A Directory

If you frequently work in one directory, but want to search other directories for files or start a program, you have to enter CD along with the complete path specification to select the old directory where you were working.

The two batch files, CD_SET.BAT and CD_GET.BAT, use an auxiliary file called CD.DAT to return to a directory.

CD_SET

```
@REM CD_SET.BAT
@CD >C:\BAT\TEMP.$$$
@COPY /B C:\BAT\CD.DAT+C:\BAT\TEMP.$$$ C:\BAT\_CD.BAT >NUL
@CALL C:\BAT\_CD.BAT
```

CD_GET

```
@REM CD_GET.BAT
@CD %CURRENTDIR%
```

CD.DAT

```
@SET CURRENTDIR=
```

The CD.DAT file on the companion diskette is set up as a read-only file, so it cannot be deleted by normal means. If you need to create your own CD.DAT file, make sure that it has only one line of text without any additional CRs or other characters, or an error message will result. This can happen if you use EDIT or a word processor to generate the CD.DAT file.

If you must create your own CD.DAT file, use COPY CON and type the data as follows:

```
COPY CON CD.DAT Enter
@SET CURRENTDIR=F6  Enter
```

The three files have to be stored in the C:\TIPTRICK directory. If you place batch files in a different directory, you have to change the CD_SET.BAT accordingly.

CD_SET stores the current directory in a temporary file. This temporary file and the auxiliary file CD.DAT create a new batch file called _CD.BAT, which places a variable in the environment space. You can use the MS-DOS SET command to view the new variable after the first time you call CD_SET.

CD_SET and CD_GET will not allow you to move between drives. For example, if you run CD_SET while in drive C:\BAT, then change to drive D:\CATS\SIAMESE, CD_SET will not return you to C:\BAT.

17.8 Quick Directory Changes

Changing directories on a hard drive can be inconvenient and time-consuming.

The following method lets you move directly to any subdirectory. Create a batch file containing the appropriate CD command for each subdirectory in your batch directory. For example, to make C:\TEXTS the current directory, use the batch file TEXTS.BAT, which contains the following line:

```
@CD C:\TEXTS
```

You could spend time creating appropriate batch files for the entire hard drive with all of its directories, or simply have a short BASIC program called MKDIRBAT.BAS perform the same task for you in seconds:

MKDIRBAT.BAS

```
REM MKDIRBAT.BAS Create batches for all directories
SHELL "CHKDSK /V|FIND " + CHR$(34) + "Directory" + CHR$(34) +
" >C:\TIPTRICK\DIR.TMP"
OPEN "C:\TIPTRICK\DIR.TMP" FOR INPUT AS #1
LINE INPUT #1, N$
WHILE NOT EOF(1)
LINE INPUT #1, N$
PATH$ = MID$(N$, 13, LEN(N$) - 5)
GOSUB MAKEADIR: REM Get Directory to Dir$
OPEN "C:\TIPTRICK\" + DIR$ + ".BAT" FOR OUTPUT AS #2
PRINT #2, "CD ";
PRINT #2, PATH$
CLOSE #2
PRINT "File "; DIR$; ".BAT"; " created."
WEND
CLOSE #1
END
MAKEADIR: REM Get Directory to Dir$
I = LEN(PATH$)
WHILE I > 1 AND MID$(PATH$, I, 1) <> "\"
I = I - 1
WEND
DIR$ = MID$(PATH$, I + 1, LEN(PATH$) - I)
RETURN
```

Before starting the program, make sure that the batch directory C:\TIPTRICK exists and make it the current directory. Start QBasic, open and run the MKDIRBAT.BAS program. It will create the desired batch files and place them in the C:\TIPTRICK directory.

This program has a number of limitations. First, it depends on the screen output of the CHKDSK command. If the program doesn't create any batch files, then CHKDSK together with FIND won't create an output file. Check the screen output of CHKDSK /V to find the correct search text for the directory lines. Second, you can create the batch files in any other directory by changing three lines in the program. Don't use the DOS directory, though, because the many batch files would clutter this directory.

Then, to change from C:\PROGRAM\WORD to C:\DATA\PERSONAL\MAIL, all you have to do is type:

```
MAIL
```

The program will run smoothly only if all subdirectories are named without extensions. Directory names such as COMPANY.BAT or DOS3.3 cannot be used here. Use the MS-DOS Shell RENAME command to remove any extensions from directory names.

Remember that these batch files for changing directories are not automatically updated when you make changes to the directory tree. If you remove a directory and then call the batch file to change to that directory, you'll get an error message. Also, you should create a new batch file for changing directories the same time that you create a new directory.

17.9 Saving Disk Space With GO.BAT

In our last section we showed you a short BASIC program that makes it easy to change directories. For each directory, the program automatically created a batch file with the name of the directory. For example, to make C:\DBASE\DATA the current directory, all you would have to do is type:

```
DATA
```

This method has one disadvantage: The batch files generated by MKDIRBAT.BAS waste a lot of space (each batch file takes up a cluster of at least 2048 bytes, regardless of its actual size). The following batch generator GO.BAS creates a single batch file called GO.BAT for all of the directories. The program passes the name of the directory to the batch as its parameter. Instead of 230,000 bytes for all those single batch files, for example, we only needed 2024 bytes, a savings of 99 percent.

After you start the program and create the batch file, it's really easy to change directories. Instead of entering CD C:\DBASE\DATA, just enter GO DATA. Although this is a little more trouble than just entering the name of the directory, the amount of space that you save makes a big difference.

GO.BAS

```
REM GO.BAS Creates GO.BAT - by Manfred Tornsdorf
REM GO.BAT enables easy change to any directory
SHELL "CHKDSK /V | FIND " + CHR$(34) + "Directory " + CHR$(34)
+ " >GO.TMP"
OPEN "GO.TMP" FOR INPUT AS #1
LINE INPUT #1, N$
OPEN "GO.BAT" FOR OUTPUT AS #2
PRINT #2, "@ECHO OFF"
PRINT #2, "SHIFT"
PRINT #2, "GOTO %0"
WHILE NOT EOF(1)
LINE INPUT #1, N$
PATH$ = MID$(N$, 13, LEN(N$) - 5)
GOSUB MAKEDIR: REM Get Directory to Dir$
PRINT #2, ":";
PRINT #2, DIR$
PRINT #2, "CD ";
PRINT #2, PATH$
PRINT #2, "Goto end"
PRINT "Entry for "; DIR$; " created."
WEND
PRINT #2, ":End"
PRINT #2, "@ECHO ON"
CLOSE #1
PRINT "To change to any directory:"
PRINT "Input: GO Dirname"
KILL "GO.TMP"
```

```
END
CLOSE #2
MAKEDIR: REM Get Directory to Dir$
I = LEN(PATH$)
WHILE I > 1 AND MID$(PATH$, I, 1) <> "\"
I = I - 1
WEND
DIR$ = MID$(PATH$, I + 1, LEN(PATH$) - I)
RETURN
```

It's best if you save this program in your TIPTRICK directory and start it from there, because this automatically places the batch file created by the program in this directory.

If you use GW-BASIC instead of QBasic to create the program (line numbers required) and your version of GW-BASIC still doesn't have the SHELL command, then type the following command line in the TIPTRICK directory of MS-DOS:

```
CHKDSK /V | FIND "Directory " >GO.TMP
```

You also have to remove the line with the SHELL command from the program.

You will have problems if there are more than one directory on the hard drive with the same name. Then you can only use GO.BAT to go to the first directory in the directory tree.

The program will only run smoothly if all subdirectories are named without extensions. Directory names such as COMPANY.BAT or DOS3.3 cannot be used here. Use the MS-DOS Shell RENAME command to remove any extensions from directory names.

The batch file GO.BAT is not automatically updated. If you remove a directory and then call the batch file to change to that directory, you'll get an error message. If you create a new directory, call the BASIC program again to create a current version of GO.BAT.

17.10 Using The <F11> And <F12> Keys

Many extended keyboards have two additional function keys, F11 and F12. Since they are seldom used by any applications, you can "program" them to add useful commands. To do this, you have to use the ANSI.SYS device driver by including the statement DEVICE=ANSI.SYS in your CONFIG.SYS file. Make certain that you use the ANSI driver of MS-DOS 5.0, since only this version of MS-DOS supports both function keys.

Now you can insert two additional lines to the AUTOEXEC.BAT, which add the DIR /W command to F11 and the DIR /P command to F12:

```
@PROMPT $e[0;133;"DIR /W";13p

@PROMPT $e[0;134;"DIR /P";13p
```

Press F11 to get a wide (multi-column) directory.

Press F12 to get a page by page directory display.

17.11 Page By Page Directory Display

The environment variable DIRCMD

Users often use the DIR command, only to have the contents of the directory rush past them on the screen. Wouldn't it be nice if the command always stopped after a page without your having to specify the /P switch? There is a way, using the environment variable DIRCMD. You can use this variable to preset parameters that are used each time you use DIR. To always display directory contents a page at a time, set the environment variable as follows:

```
SET DIRCMD=/P
```

You can specify all of the parameters that the DIR command allows for DIRCMD. The following displays directories one page at a time, and in alphabetical order:

```
SET DIRCMD=/O:N /P
```

DIRCMD parameters are lost when you switch off the computer. To restore these parameters, you can enter a command the next time you switch on your computer. Alternatively, you can add DIRCMD to your AUTOEXEC.BAT file, so that the variable is automatically set each time you start the computer.

17.12 Protecting Directories

Did you ever wish you could keep people from viewing files in a certain directory? There is an easy method of protecting a directory. Although this method does not offer complete protection, it can keep the casual user from accessing the directory. You can choose directory names that can make it difficult for unauthorized users to access the directory.

Just choose a directory name that they can't enter easily. The hidden space (ASCII value 255) is an especially interesting option. It looks like an ordinary space (ASCII value 32), but is not a valid character for MS-DOS.

Here's what you would type to protect a new directory:

```
MD NAME Alt + 2 5 5
```

For Alt + 2 5 5, just hold down the Alt key and type 2 5 5 on the numeric keypad. Now, if someone tries to enter the directory name, they probably won't enter the hidden space. Since they aren't entering the full directory name, they won't be able to access the directory. To access the protected directory, type:

```
CD NAME Alt + 2 5 5
```

You must remember which directories are protected in this manner or you won't be able to access the directories either.

17.13 Cancelling Disk Formatting

When you format a hard drive, the data is not destroyed or changed at first. Instead, FORMAT first checks all of the sectors of the hard drive for readability. As it's doing that, it displays the head and cylinder information.

When this is complete, it does change the contents of the hard drive by writing a new, empty root directory and corresponding FAT (File Allocation Table) to the hard drive.

This means that you can cancel FORMAT to a hard drive without losing data while FORMAT is still checking the sectors of the hard drive. Press Ctrl + C, or switch off your computer.

Using the MS-DOS UNFORMAT command, you can undo the formatting in many cases, but it's better yet to abort the formatting process.

We hope you don't have to use this method unless you have good reason to do so.

17.14 Testing Files For Readability

Have you ever copied an important file from the hard drive onto a diskette only to discover later that you can't read the file correctly?

Or have you ever gotten an error message from an application while editing a file and you didn't know whether it was because you weren't able to read the file?

It's easy to find out whether diskettes contain defects by using DISKCOPY. If MS-DOS cannot copy parts of the diskette correctly, it displays an error message describing the read errors. You can't use DISKCOPY when you're using the hard drive, though.

You could use COPY to copy the file, but this takes some time, and then you have a copy of the file that you really don't need. MS-DOS provides a special device, NUL, that "absorbs" all of the incoming data. This gives you an extremely quick option for testing files for readability:

```
COPY File NUL
```

The dummy device NUL

MS-DOS either reports that it copied a file, or that a read error occurred. You check more than one file by using the wildcards "?" and "*". To check all of the files in drive A: for readability, use the following command:

```
COPY A:*.* NUL
```

17.15 Contiguous Files

The longer you work with a diskette or hard drive, the more the files spread out as non-contiguous (scattered) data. Because of this, it takes longer and longer to access data in the file. You can read files more quickly if the data is contiguous.

Optimal storage of files

To check the distribution of a file on a storage medium, you can use the following command:

```
CHKDSK filename
```

CHKDSK gives the usual information about the storage medium and memory in the PC and tells the relative efficiency of a file:

```
All specified files are contiguous
The file is optimally allocated
Filename contains X non-contiguous blocks
File is not contiguous, but consists of X fragments
```

You can also use wildcards for a filename. If a disk contains many non-contiguous files, use BACKUP to back up all of the data, format the disk and then use RESTORE to recover the data. It's a lot quicker if you use special utility programs to organize a disk (e.g., *BeckerTools Compress* from Abacus).

17.16 Disabling <Num Lock>

Some versions of BIOS (especially on PC/ATs) automatically turn on Num Lock when the computer is first started. This prevents you from entering anything besides numbers using the numeric keypad - the cursor control maintained by these keys is disabled. To use the numeric keypad for cursor control, you have to deactivate Num Lock by pressing the Num Lock key.

Here is a short assembly language program that changes the status of Num Lock each time you reset. The PC's Num Lock LED will remain unchanged (the status of Num Lock will change), but the LEDs on AT and 386 keyboards will change when you call this program.

Type the following program like a batch file, and then save it under the name NUMLOCK.DEB.

NUMLOCK.DEB

```
A
MOV AX,0040
MOV DS,AX
MOV AL,[0017]
XOR AL,20
MOV [0017],AL
RET

RCX
000E
NNUMLOCK.COM
W
Q
```

The blank line between RET and RCX is important.

Now use DEBUG to transform the commands into a machine language program. Type the following line:

```
DEBUG <NUMLOCK.DEB
```

Several lines appear on the screen with the following message at the end:

```
Writing 000E bytes
-Q
```

DEBUG generates the finished program NUMLOCK.COM, which switches between cursor control and numeric input each time you run it. If cursor control is deactivated after each reset, you can copy the program to your DOS directory and insert the command in your AUTOEXEC.BAT file, as you would for any other MS-DOS command.

17.17 Skipping Commands In AUTOEXEC.BAT

As you find new and better ways to use your PC, you may find that you're adding new commands and programs to your AUTOEXEC.BAT file. You may not even realize that most of these conveniences require memory. So if you run one of your applications, you may encounter a message such as "Not enough memory."

You might add a short program to your AUTOEXEC.BAT file that asks you if you want to execute a particular command during startup. But this defeats the purpose of an automatic file.

The following program KBFLAG.COM, which is used from an AUTOEXEC.BAT file, gives you the option of skipping parts of AUTOEXEC.BAT. To do this, the program checks whether Ctrl is being pressed and returns a message to AUTOEXEC.BAT, so that the remainder of the commands in the AUTOEXEC.BAT file can be skipped.

To use this program, create it as described and modify your AUTOEXEC.BAT by adding a few new lines. Also, all of the additional programs that aren't necessary should be in one section (e.g., at the end of the file) so that it's easy to skip them.

Type the following program like a batch file, and then save it under the name KBFLAG.DEB.

KBFLAG.DEB

```
A
MOV AX,40
MOV DS,AX
MOV AL,[17]
AND AL,04
MOV AH,4C
INT 21

RCX
E
N KBFLAG.COM
W
Q
```

The blank line between INT 21 and RCX is important.

Transform the commands to a machine language program. To do this, type the following line:

```
DEBUG <KBFLAG.DEB
```

Now you have a finished program called KBFLAG.COM. When you call it from AUTOEXEC.BAT, you get an ERRORLEVEL 4 when you hold down Ctrl while the program is executing. IF ERRORLEVEL 4 GOTO ABORT lets the batch file jump to the end.

Num Lock has to be turned off before running KBFLAG.COM. If necessary, call NUMLOCK.COM before calling KBFLAG.COM. Important system programs such as keyboard layout and path specifications should be finished before the batch uses KBFLAG.COM.

To use the KBFLAG.COM program properly, add the following lines ahead of the commands that you want to skip:

```
REM The statements before this line are the ones that you
REM always want to execute from AUTOEXEC.BAT
REM --------------------------------
NUMLOCK
KBFLAG
IF ERRORLEVEL 4 GOTO Abort
REM --------------------------------
REM The statements here can be skipped if
REM the Ctrl key is pressed
ECHO Additional commands go here
GOTO END
:Abort
ECHO Keyboard CTRL pressed!
ECHO Additional programs not installed
:END
REM AUTOEXEC.BAT ends here
```

If you don't press Ctrl while AUTOEXEC.BAT is running, the entire AUTOEXEC.BAT file is run as usual.

If you hold down Ctrl while the batch file is running, however, the following appears:

```
Keyboard CTRL pressed!
Additional programs not installed
```

This provides you with the flexibility to conserve memory for applications that require it occasionally.

17.18 Determining Disk Space

Have you ever wondered how much space a group of files requires on a diskette or hard drive? This is especially important when you want to copy all of the files from a directory of the hard drive onto a diskette, and you need to know whether there is enough room for them. Or perhaps you would like to know how many diskettes you need for a backup.

The DOS DIR command displays the amount of space that the files in a directory use. But since this DIR command displays the value, you can't automatically use this information in a program.

The following QBasic or GW-BASIC program solves this problem:

DISPLBYT.BAS

```
10 REM This program determines the size of files saved in
20 REM DIR.TXT
30 SHELL "DIR >$$$$$$$$.TMP"
40 CLS
50 DISPLBYT = 0
60 OPEN "$$$$$$$$.TMP" FOR INPUT AS #1
70 WHILE NOT EOF(1)
80 LINE INPUT #1,FILE$
90 IF LEFT$(FILE$,8)<> "$$$$$$$$" THEN DISPLBYT = DISPLBYT +
VAL (MID$(FILE$,13,10))
100 PRINT ".";
110 WEND
120 PRINT
130 PRINT "Memory used by files: ",DISPLBYT - 11
140 CLOSE #1
150 KILL "$$$$$$$$.TMP"
160 SYSTEM
```

You can omit the line numbers in QBasic.

Explanation

The program uses the QBasic SHELL command to start an MS-DOS command from BASIC. DIR is used as a command and the screen output of the DIR command is redirected to a file. The filename used in the program is somewhat strange, but prevents an important file with the same name from being accidentally overwritten.

Next, the program reads the contents of the file, line by line, until the end of the file. The program isolates the values containing the file size from each line. The file sizes are totalled and the size of the temporary file containing the directory output ($$$$$$$$.TMP) is subtracted from the calculation. Finally, the file is closed and the results are displayed on the screen.

The information from DIR about the file size indicates the size of the file, but not the amount of space that the file takes up on the diskette or hard drive. This amount depends on the minimum amount of memory MS-DOS uses for a file. For a 360K diskette, the minimum is 1024 bytes, for a 1.2 Meg the minimum is 512 bytes, and for a hard drive on a PC-AT it's usually 2048 bytes.

17.19 Fast Keyboard

Perhaps you have an AT or even a 386, but your cursor seems to crawl across the screen when you type characters at the command interpreter. Here's a possible solution to your problem.

Use the MODE command. One parameter specifies the speed at which the cursor moves across the screen. The other parameter specifies the delay time - the time a key has to be held down before automatic repetition of a pressed key starts. Here's what the MODE command looks like:

```
MODE CON RATE=<speed> DELAY=<delay>
```

<speed> ranges from 1 (slow) to 32 (fast). <delay> ranges from 1 to 4, in increments of 1/4 second.

To get the highest possible speed from your keyboard, type the following command:

```
MODE CON RATE=32 DELAY=1
```

When you try to change the speed and delay of the keyboard, you may get the message:

```
Function not supported on this computer
```

This means that not even MS-DOS can do anything about the situation. You won't be able to configure your keyboard.

To permanently change the keyboard speed, insert the command in your AUTOEXEC.BAT file.

17.20 Expanding The Path Specification

You're already familiar with the PATH command. It defines the order in which directories are searched to locate executable program or batch files. But adding directories to an existing PATH command usually requires that you retype the existing command and add the new directory to the end.

The following batch file provides a remedy. We call it ADDPATH.BAT:

ADDPATH.BAT

```
@ECHO OFF
PATH %PATH%;%1
ECHO New path: %PATH%
@ECHO ON
```

When using this batch file, specify the new directory which you want to add to the search path. After this, the new, complete path specification is displayed. If your original path specification contains only C:\DOS, for example, and you want to add C:\BATCH, use the following line:

```
ADDPATH C:\BATCH
```

Then the screen displays the message:

```
New path: C:\DOS;C:\BATCH
```

The next time you turn on the computer, the extended search path is no longer valid. If you want to make the path extension permanent, expand the path by adding the new path to the PATH command in your AUTOEXEC.BAT file.

17.21 Copying Groups Of Files With GCOPY

Although you can use COPY to copy more than one file by using wildcards in the command, this assumes that there is some commonalty in the filenames. You can't copy all files with both .TXT and .BAK extensions, for example.

The batch file GCOPY.BAT lets you copy groups of files with a single command:

GCOPY.BAT

```
@ECHO OFF
IF "%1"=="" GOTO END
IF "%2"=="" GOTO END
SET TARGET=%1
SET NAME=%0
:GROUP
IF "%2"=="" GOTO END
ECHO %NAME% COPYING %2 TO %TARGET%
COPY %2 %TARGET%
SHIFT
GOTO GROUP
:END
SET TARGET=
SET NAME=
```

Enter the target (drive and path, just like the COPY command) as your first parameter after the batch file and then the desired filenames with or without wildcards.

The following copies all of the files with .TXT or .BAK extensions residing in the current directory to drive A:

```
GCOPY A: *.TXT *.BAK
```

☞ The "%0" parameter contains the name of the batch file during the call.

17.22 Excluding Files From Commands

Let's say that you want to back up the data from the DOCUMENT directory to a diskette. This directory contains all of your document files, graphics, style sheets and glossary files required by the document files.

However, your word processor generates a BAK file for each document file, so the DOCUMENT directory also contains BAK files. You don't want to copy the BAK files, otherwise there won't be enough room on the diskette for all the other files. On the other hand, you don't want to delete the backup copies from your hard drive.

Or let's say that you want to delete all of the files in a directory except the document files. You can do this using MS-DOS 5.0 and the following batch file.

NOT.BAT only works with MS-DOS 5.0, and for commands that refer to the current directory.

NOT.BAT

```
@ECHO OFF
@REM ╔══════════════════════════════════════╗
@REM ║ NOT.BAT  Excludes certain files      ║
@REM ╚══════════════════════════════════════╝
IF "%1" == "" GOTO Syntax
IF "%2" == "" GOTO Syntax
IF NOT EXIST %2 GOTO EXISTNOT
ATTRIB +H %2 > NUL
%1 *.* %3 %4 %5 %6 %7 %8 %9
ATTRIB -H %2 > NUL
GOTO END
:EXISTNOT
IF NOT EXIST %2*.* GOTO DirTest
CALL NOT.BAT %1 %2*.* %3
GOTO END
:DirTest
IF NOT EXIST %2\*.* GOTO ISNOT
CALL NOT.BAT %1 %2\*.* %3
GOTO END
:ISNOT
ECHO.
ECHO ╔══════════════════════════════════════════╗
ECHO ║     The file(s) that you specified don't ║
ECHO ║        exist in this directory anyway!   ║
ECHO ╚══════════════════════════════════════════╝
GOTO END
:SYNTAX
CLS
ECHO.
```

```
ECHO ╔═══════════════════════════════════════════════╗
ECHO ║ NOT.BAT executes an MS-DOS command from which ║
ECHO ║ the specified files are excluded!             ║
ECHO ║ --------------------------------------------- ║
ECHO ║ Syntax: NOT DOS_command filename parameters   ║
ECHO ║ DOS_Command: e.g., COPY or DEL or XCOPY       ║
ECHO ║ Filename:    name or group of files that      ║
ECHO ║              are to be excluded: e.g.,        ║
ECHO ║              TEXT.BAK or *.TMP or TEX*.*      ║
ECHO ║ Parameters:  any necessary parameters         ║
ECHO ║              such as A: B: /P /S etc.         ║
ECHO ╚═══════════════════════════════════════════════╝
:END
@ECHO ON
```

The batch file NOT.BAT takes advantage of special options of the ATTRIB command in MS-DOS 5.0. Using Version 4.0 and earlier you could only use ATTRIB to change the Archive and Read-Only file attributes. With MS-DOS 5.0, you can also change the Hidden and System attributes. If a file has the hidden attribute, it is "invisible" for most commands, that is, it is ignored by these commands. That's just what we need for the special function used in NOT.BAT.

How it works

The batch file NOT.BAT works in the following manner: First, all of the files specified in the file selection (second parameter) are hidden with the ATTRIB command (ATTRIB +H %2 > NUL). The redirection to the dummy device NUL redirects messages from the ATTRIB command away from the screen.

Then the specified MS-DOS command (first parameter) can be used for all files (*.*). Finally, all of the files from the excluded file group (second parameter) are made visible again (i.e., the hidden file attribute is removed). To copy all files in a directory except for the backup copies, you type:

```
NOT COPY *.BAK A:
```

This line means: COPY ALL FILES EXCEPT *.BAK TO A:.

You could also use NOT.BAT together with the MS-DOS DEL command (after careful consideration). Let's say you have a directory with many files (documents, glossaries, style sheets, backup files, etc.) and it turns out that you only want to keep the DOC files - you want to delete the other files for reasons of space.

It's no problem with our new batch file NOT.BAT. If the directory is the current MS-DOS directory, it's enough to use the following command:

```
NOT DEL *.DOC
```

Don't be surprised if you encounter a prompt for the DEL command, because the batch file uses the DEL *.* - that's where the prompt comes from. The document files were made invisible beforehand, though.

When you use the batch file NOT.BAT, an unwelcome side effect could pop up if there were already hidden files in the current directory beforehand. Let's say that you exclude EXE files and there are two hidden EXE files in the current directory. While all of the EXE files would be excluded from the specified MS-DOS command, they would also be visible.

Although you can completely avoid this effect (and exclude previously hidden files completely from this process), the batch file necessary for this would be several pages long. That's why we limit ourselves to a simpler solution, because the problem probably won't come up too often.

We display all of the files in the batch file beforehand, that would be affected by this (unintentional) change. That way, you can hide these files again afterwards, if necessary. To do this, simply insert between the lines:

```
IF NOT EXIST %2 GOTO EXISTNOT

ATTRIB +H %2 > NUL
```

the following new lines:

```
CLS
ECHO.
ECHO ╔══════════════════════════════════════════════╗
ECHO ║                   WARNING!                   ║
ECHO ║  Some of the files that should NOT be        ║
ECHO ║ processed might be protected as "hidden" files.║
ECHO ║ If one or more files are displayed after this║
ECHO ║ message they will NO LONGER be protected with║
ECHO ║ this hidden attribute afterwards!            ║
ECHO ║      Please press any key...                 ║
ECHO ║                                              ║
ECHO ╚══════════════════════════════════════════════╝
ECHO.
PAUSE >NUL
DIR %2 /A:H
ECHO.
ECHO ╔══════════════════════════════════════════════╗
ECHO ║      Please note the filenames if necessary  ║
ECHO ║    and use ATTRIB to set the file attribute  ║
ECHO ║              Hidden.                         ║
ECHO ║      please press any key...                 ║
ECHO ║                  ABORT with CTRL + C         ║
ECHO ╚══════════════════════════════════════════════╝
```

```
ECHO.
PAUSE >NUL
```

The companion diskette contains a batch file named NOT2.BAT which includes these updated lines and the NOT.BAT file combined.

If hidden files are displayed by the batch file, you can write down the names. After the batch file is finished running, you can reset the Hidden attribute for these files.

Let's say that the COMMAND.EXE file was hidden beforehand. Then you called the batch file with NOT DEL *.EXE. The COMMAND.EXE file would be displayed at the beginning as a hidden file and at the end of the batch you could use the following command to hide it again:

```
ATTRIB COMMAND.EXE +H
```

17.23 Warning Beep In Batch Files

As you create powerful, efficient batch files, you may want to warn the user about errors. An example of such a warning beep is if you enter more than 127 characters in a command line.

You can create a warning beep by typing the following command line:

```
@ECHO Ctrl + G
```

Hold down the Ctrl key and press G. The screen displays the key combination as ^G.

So that you don't have to memorize this key combination, create a short batch file called BEEP.BAT, that contains this line. Don't forget to place the "@" character in front of the ECHO command, or ECHO will be displayed on the screen. Then you can create a warning beep from a batch file with the following command line:

```
CALL BEEP
```

The CALL command ensures that the batch file continues running after you call BEEP. In some word processing programs (such as Microsoft Word), you cannot create the warning beep by pressing Ctrl + G. In that case, use COPY CON BEEP.BAT and end the command after entering the line with Ctrl + Z.

If you use TYPE to output BEEP.BAT on the screen, you will also hear the warning beep, because the MS-DOS TYPE command outputs Ctrl + G just like ECHO, and MS-DOS converts it to the warning beep.

17.24 Customizing PAUSE

The PAUSE command will not let you enter your own prompt. PAUSE defaults to the message, "Press any key to continue...".

Here's a simple trick to get around this message: A batch file called MESSAGE.BAT, which suppresses the default PAUSE message and inserts your own message in its place:

MESSAGE.BAT

```
@ECHO %1 %2 %3 %4 %5 %6 %7 %8 %9
@PAUSE >NUL
```

To have the program display any message on the screen and then wait for the press of a key, include the following message in a batch file:

```
CALL MESSAGE Your message written here up to nine words total
```

The actual text of the message can be up to nine words.

17.25 Trace Mode For Batch Files

When you start writing batch files, you're bound to encounter a few problems just as you would in any programming language. This is especially true when you are developing large batch files, when errors are very difficult to locate.

Usually a batch file starts with the command ECHO OFF, and once that occurs you can't figure out which part of the batch file led to the error message.

Finding errors in batch files

You can use a simple trick to implement a trace mode for testing batch files, that is, you can have the batch display the command lines on the screen before they are executed. To do this, define a new environment variable, DEBUG.

Write the desired status of ECHO in the variable. To have the batch file run without displaying lines, type the following command (it's best if you type it in the AUTOEXEC.BAT):

```
SET DEBUG=OFF
```

To find an error in a batch file, just type the following command line immediately before calling the batch file:

```
SET DEBUG=ON
```

To use this technique, you just have to make a simple change to the batch files you're testing. Wherever ECHO OFF would ordinarily be written, the new line now reads ECHO %DEBUG%. MS-DOS then automatically inserts either ON or OFF, depending on the setting of the DEBUG environment variable.

If you start such a batch file without using SET to define the contents of the DEBUG variable, then the actual status of ECHO is displayed. After that, MS-DOS lists each batch line before executing it.

The following batch file, BUGS.BAT, contains an error and is already running with trace mode. Test the program by entering SET DEBUG=OFF before calling the batch file. The batch file is executed, but doesn't deliver the desired result. You could enter "Work" as a parameter (without quotation marks), but the label "Work" still won't be called.

BUGS.BAT

```
@ECHO %DEBUG%
REM This batch file contains an error
REM Use SET DEBUG=ON for display
REM Display can be prevented by SET DEBUG=OFF
```

```
IF "%1" == Work GOTO
Goto End
:Work
Echo There is work here
ECHO Abort with CTRL + C
GOTO Work
:End
@ECHO ON
```

To find the error, just enter SET DEBUG=ON before calling it again. When you call the batch file after setting DEBUG to ON, each line appears on the screen as executed in the batch file.

You'll then see that the comparison is wrong. Additional quotation marks should be inserted after the double equal sign, and the word Work should follow the GOTO statement.

17.26 Sorting Files In A Path

FOR...DO can be an extremely powerful command in batch files, which you can use to do a lot more than just program loops and trace any number of files in a directory. We want to use an interesting example to show you what the command can do.

The following batch file, SHOWPATH.BAT, displays all of the files in the path in alphabetical order, as well as their positions in the search path. This enables you to easily determine whether you have several programs with the same name in different directories and whether there are too many files in the search path, which could result in a noticeable slowdown of MS-DOS.

SHOWPATH.BAT

```
@ECHO OFF
REM SHOWPATH.BAT - by Manfred Tornsdorf
SET OLDPROMPT=%PROMPT%
@IF EXIST SHOWPATH.TMP DEL SHOWPATH.TMP
FOR %%a in (%PATH%) DO DIR %%a >>SHOWPATH.TMP
FIND /V "." SHOWPATH.TMP | FIND /V "." | FIND /V "e" |
   FIND /V "---------" | SORT | MORE
FIND /C ":" SHOWPATH.TMP
@ECHO ON
@prompt $e[3A

@ECHO Total  files  available
@SET PROMPT=%OLDPROMPT%
@SET OLDPROMPT=
```

If the total search path is longer than one screen line (display after PATH), you may get an error message, because MS-DOS only allows command lines of up to 127 characters.

The batch file uses a number of tricks that you might use in your own batch files:

1. The contents of the environment variable PATH are inserted directly within the parentheses of the FOR...DO loop. Since the individual directories are separated by semicolons, the variable "%%a" accepts all of the directory specifications of the search path, one after the other. The DIR command displays the contents and directs them to the SHOWPATH.TMP file.
2. The double redirection character ">>" sees to it that the file is not recreated after each time DIR is called, adding the new lines to the existing file instead.

You can use this trick whenever you want to direct the screen output of several commands to a common file. You must delete an existing file with this name beforehand. The condition IF EXIST... causes this file to be deleted only if it exists.

3. After all of the filenames and these paths are written to the temporary file, the FIND command filters out all unwanted elements at multiple levels (information about the storage medium, the special entries "." and ".." and a special display of the FIND command (---------- Filename).

4. The batch file uses two techniques to determine and then output the number of files. First the FIND /C "." command counts all of the lines with a period. These are the files that have to do with the date and time. Here's what the output of the FIND command looks like:

   ```
   ---------- SHOWPATH.TMP: 345
   ```

   A special ESC sequence for ANSI.SYS places the cursor three lines above using the PROMPT command. Then the ECHO command uses "Total files available" to overwrite the file specification of the FIND command, so that the number of lines appears directly after the default text.

   By the way, the blank line after the PROMPT command is extremely important for the operation, and cannot be omitted or suppressed by ECHO OFF, since this blank line causes the PROMPT command to go into effect exactly one time.

5. When you make a change to PROMPT, a batch should then restore the original ready prompt. For this purpose, it is saved in the new environment variable OLDPROMPT at the start, and then reset at the end. After this, the variable OLDPROMPT is removed by a blank instruction.

This batch file only runs correctly if the complete path specification is not too long. You can experiment by calling the MS-DOS PATH command. A search path that is 1-1/2 screen lines long results in a syntax error, because the entire command line is too long.

17.27 Keyboard Shortcuts - SETFUNC0

The function keys F1 to F10 serve only very limited purposes at the MS-DOS command line. The most useful key is F3, since you can use it to repeat the last command line.

In principle, you can use ESC sequences to program the keyboard to the device driver, ANSI.SYS. Since the commands are awkward and complicated, the following batch file SETFUNC0.BAT allows you to assign actions to the function keys easily. Remember that the last character of the filename is a "zero" rather than a letter "O."

SETFUNC0.BAT

```
@ECHO OFF
REM SETFUNCO.BAT - by Manfred Tornsdorf
REM Function key setup
REM -----------------------------------------------------    BEGIN
SET OLDPROMPT=%PROMPT%
IF "%1" == "" GOTO Syntax
GOTO Start
REM -----------------------------------------------------   SYNTAX
:Syntax
ECHO.
ECHO You must enter at least one function key.
ECHO.
ECHO SETFUNCO.BAT allows you to assign tasks to function keys.
ECHO =============================================================
ECHO = Syntax: SETFUNCO FUNCTION-KEY [Assignment] [/ENTER]       =
ECHO =    Enter FUNCTION-KEY parameter in upper case.            =
ECHO =    Entering FUNCTION-KEY without [Assignment] deletes     =
ECHO =    the current assignment.                                =
ECHO =                                                           =
ECHO =    [/ENTER] inserts the Enter key.                        =
ECHO =                                                           =
ECHO =    Examples:                                              =
ECHO =    SETFUNCO F3 : Deletes current assignment from F3       =
ECHO =    SETFUNCO F3 DIR : Assigns DIR to F3                    =
ECHO =    SETFUNCO F3 DIR /ENTER: Assigns DIR {Enter} to F3      =
ECHO =============================================================
ECHO.
GOTO EndItAll

REM--------------------------------------------------------- Start
:Start
SET FuncKey=%1
IF "%2" == "" GOTO Functest
:TheLoop
IF "%2" == "/ENTER" GOTO EnterChar
IF NOT "%Allocation%"=="" GOTO N_Blank
SET Allocation=%2
GOTO Continue
```

```
REM-------------------------------------------------------- N_Blank
:N_Blank
SET Allocation=%Allocation% %2
:Continue
SHIFT
IF NOT "%2" == "" GOTO TheLoop
GOTO FuncTest

REM------------------------------------------------------- EnterChar
:EnterChar
SET Enter=;13
SET EndChar=/Enter
GOTO FuncTest
REM-------------------------------------------------------- FuncTest
:FuncTest
IF "%FuncKey%" == "F1" SET Fnum=59
IF "%FuncKey%" == "F2" SET Fnum=60
IF "%FuncKey%" == "F3" SET Fnum=61
IF "%FuncKey%" == "F4" SET Fnum=62
IF "%FuncKey%" == "F5" SET Fnum=63
IF "%FuncKey%" == "F6" SET Fnum=64
IF "%FuncKey%" == "F7" SET Fnum=65
IF "%FuncKey%" == "F8" SET Fnum=66
IF "%FuncKey%" == "F9" SET Fnum=67
IF "%FuncKey%" == "F10" SET Fnum=68
IF "%Fnum%" == "" GOTO Syntax
GOTO Allocation
REM------------------------------------------------------ Allocation
:Allocation
IF "%Allocation%" == "" GOTO RemoveAssgnmt
@ECHO ON
@Prompt $e[0;%Fnum%;"%Allocation%"%Enter%p

@ECHO OFF
GOTO EndItAll
REM--------------------------------------------------- RemoveAssgnmt
:RemoveAssgnmt
@ECHO ON
@PROMPT $e[0;%Fnum%;0;%Fnum%p

@prompt $e[2A

@ECHO OFF
REM-------------------------------------------------------- EndItAll
:EndItAll
SET %FuncKey%-KEY=%Allocation%%EndChar%
SET Fnum=
SET Allocation=
SET FuncKey=
SET Enter=
SET EndChar=
@SET PROMPT=%OLDPROMPT%
@SET OLDPROMPT=
```

To cancel this key action and return the original function to F2 (copy up to specified character), type the following line:

```
SETFUNC0 F2
```

You have to specify the function keys in capital letters when you call SETFUNC0.BAT. Entering SETFUNC0 F2 DIR will work, whereas SETFUNC0 f2 DIR will not. You must also specify the /ENTER parameter in capital letters. Entering SETFUNC0 F2 DIR /ENTER will work, whereas SETFUNC0 F2 DIR /Enter will only include the "/Enter" as part of the command proper, instead of executing the command immediately. We recommend that you enter the command in all capitals.

If you write a short batch file called HELPSCRN.BAT that outputs useful information and help on the screen, you could assign the help function to F1, as is the case with most professional programs. Use the following command line:

```
SETFUNC0 F1 HELPSCRN /ENTER
```

The companion diskette contains a batch file named SETFUNC.BAT. SETFUNC.BAT is a more advanced version of this program. See Appendix E for more information about the companion diskette.

17.28 Displaying Function Key Assignments

If you use SETFUNC0.BAT to assign an action to a function key, this action is automatically saved in an environment variable. You can check this easily, by entering the SET command after the system prompt.

Each action is saved in the following form:

```
Fx-Key=Action
```

The following batch file, GETFUNC0.BAT, uses this fact to display the current function key assignments:

GETFUNC0.BAT

```
@REM GETFUNC0.BAT - by Manfred Tornsdorf
@REM Displays current function key layout
@SET | Find "-KEY" | SORT
```

17.29 Changing Directories

The following short batch file, SAVEDIR.BAT, stores the current directory path in F10. If you run SAVEDIR.BAT, change directories and want to immediately return to the directory placed in F10, just press F10.

Enter SAVEDIR.BAT and save it in your TIPTRICK directory. Do not omit the blank lines after the two PROMPT lines.

SAVEDIR.BAT

```
@REM SAVEDIR.BAT
@REM Stores current directory in F10 for fast CD
@SET OLDPROMPT=%PROMPT%
@PROMPT $e[0;68;"CD ";"$p";13p

@PROMPT $e[5A

@SET PROMPT=%OLDPROMPT%
@SET OLDPROMPT=
```

Use the batch file, SETFUNC0.BAT, to assign this short program to F9:

```
SETFUNC0 F9 SAVEDIR /ENTER
```

Now when you want to record the current directory (e.g., C:\DATA\TEXTS), simply press F9. Then you can change directories as usual, for example, you could enter CD C:\DOS. To go back to the saved directory, just press F10. Then you are back in the C:\DATA\TEXTS directory.

Since the function key assignments are lost every time you reboot the computer, you may prefer to define the SAVEDIR call with a function key in AUTOEXEC.BAT. To assign SAVEDIR to F9, insert the following lines in AUTOEXEC.BAT:

```
@ECHO ON
@PROMPT $e[0;67;"SAVEDIR";13p

@ECHO OFF
```

17.30 Deleting Protected Files

Using DEL *.* doesn't always guarantee success. For example, if you use this command on a backup diskette made by the MS-DOS BACKUP command, the message "Access denied" appears and the files remain on the diskette. You cannot use the diskette for data storage since there is no free disk space.

With a diskette, you could use the FORMAT command, but some applications protect themselves from deletion by having read-only status applied to them.

Simple solution

The batch file KILL.BAT provides a remedy for this problem. KILL.BAT removes the protection from deletion after a security prompt and deletes the specified file(s).

KILL.BAT

```
@ECHO OFF
@REM KILL.BAT Simple version - Manfred Tornsdorf
ECHO Ready to delete %1
PAUSE
ATTRIB -R %1
DEL %1
```

The call to delete a file in the WORD directory named SAMPLE.EXE would be:

```
KILL C:\WORD\SAMPLE.EXE
```

The following line deletes all of the files in the root directory of drive A: (confirm the security prompt by typing Y):

```
KILL A:\*.*
```

Friendlier version of KILL.BAT

The KILL2.BAT file listed below is a lot more extensive, but also friendlier and more flexible toward errors.

KILL2.BAT

```
@ECHO OFF
REM KILL2.BAT
REM Removes read-only status and deletes files
IF "%1" == "" GOTO Syntax
IF NOT "%2" == "" GOTO Syntax
IF NOT EXIST %1 GOTO Not_There
ECHO KILL2.BAT ready to delete the following files:
DIR %1 | MORE
ECHO
ECHO Press CTRL + C to cancel, any other key to execute
PAUSE >NUL
ATTRIB -R %1
```

```
ECHO Y|DEL %1 >NUL
ECHO Delete successful
GOTO EndItAll
:Not_There
IF NOT EXIST %1*.* GOTO Test_For_Dir
CALL KILL2.BAT %1*.*
GOTO EndItAll
:Test_For_Dir
IF NOT EXIST %1\*.* GOTO Error_Trap
CALL KILL2.BAT %1\*.*
GOTO EndItAll
:Error_Trap
ECHO %1 does not exist
GOTO EndItAll
:Syntax
ECHO KILL2.BAT removes read-only status and deletes files
ECHO Syntax: KILL2 Filename (wildcards also allowed)
:EndItAll
@ECHO ON
```

First, the batch file checks for the correct number of parameters. If there are no parameters or more than one parameter is specified, a message describes the proper syntax, and the batch file ends. Otherwise, the batch file continues.

Then the batch command EXIST checks whether the specified file(s) even exist. If not, the batch file branches to the "Not_there" routine (more on this later).

Otherwise, the batch outputs a warning and displays all of the files that would be affected by the deletion. The PAUSE command allows you to continue by pressing a key or else abort by pressing Ctrl + C. The message of the PAUSE command is redirected to the dummy device, NUL, since you only need to wait for the press of a key and the rest of the message would only get in the way.

The dummy device NUL

After you press a key, ATTRIB removes the protection from deletion, and then all of the files are deleted. While the files are being deleted, the output on the screen is redirected to NUL, which makes it invisible. The ECHO command also automatically answers any possible security prompts (such as *.*). The batch file ends after displaying a message about the success of the deletion.

The "Not_There" routine is clever. It intercepts some special situations and allows the batch to run correctly. For example, the parameter "A:" would abort the program, since EXIST doesn't find a file, while DEL A: results in the error message "File not found". The same thing holds true for the call with a subdirectory (e.g. the parameter "A:\DATA").

That's why the "Not_There" routine checks whether a successful deletion could follow from "*.*" or "*.*" and then calls itself (calls KILL2.BAT a second time) with the improved parameters. The user doesn't even notice this process. The "Not_There" routine allows for a certain margin of error.

17.31 Batch File Prompts

It's not always possible to pass all of the necessary parameters to a batch file as parameters. Sometimes you'll need to prompt the user for additional information. Moreover, you'll sometimes need confirmation for performing a command.

Here's a short assembly language program which prompts the user for a confirmation. You don't need any special knowledge to create this program - all you need is an editor (such as EDIT), DEBUG (which comes with DOS) and the ability to type.

Use an editor to type the following lines under the filename ASK.DEB:

ASK.DEB

```
A
MOV AX,0C07
INT 21
MOV AH,4C
INT 21

RCX
9
nASK.COM
W
Q
```

The blank line between INT 21 and RCX is important. Without this line, calling DEBUG results in an error.

After saving the file, type the following line to run the MS-DOS DEBUG command:

```
DEBUG <Ask.DEB
```

Now you have a small executable program called ASK.COM. The following sample program, PROMPTME.BAT, shows you how to use ASK.COM. This batch file only works with lowercase characters:

PROMPTME.BAT

```
@ECHO OFF
REM Promptme.BAT - Batch file for use with ASK.COM
:UserInput
ECHO (y)es or (n)o?
ASK
IF ERRORLEVEL 121 GOTO Yes
IF ERRORLEVEL 110 GOTO No
GOTO UserInput
```

```
:Yes
IF ERRORLEVEL 122 GOTO UserInput
ECHO Yes
GOTO EndItAll
:No
IF ERRORLEVEL 111 GOTO UserInput
ECHO No
GOTO EndItAll
:EndItAll
@ECHO ON
```

Instead of "y" and "n", you can also prompt for other characters by using the appropriate ASCII code after ERRORLEVEL. However, you have to prompt for the ERRORLEVEL in descending sequence.

Explanation

The assembly language program that you created with DEBUG waits for you to type a character and sends back the ASCII value of the character as an error code. Within the batch file you can look for this value through ERRORLEVEL.

The condition is satisfied if ERRORLEVEL is greater than or equal to the value specified after it. That's why you have to check the highest value first. The batch program, in other words, checks in descending sequence whether the return value was larger than or equal to the value of the character to be checked and uses GOTO to branch to the appropriate place.

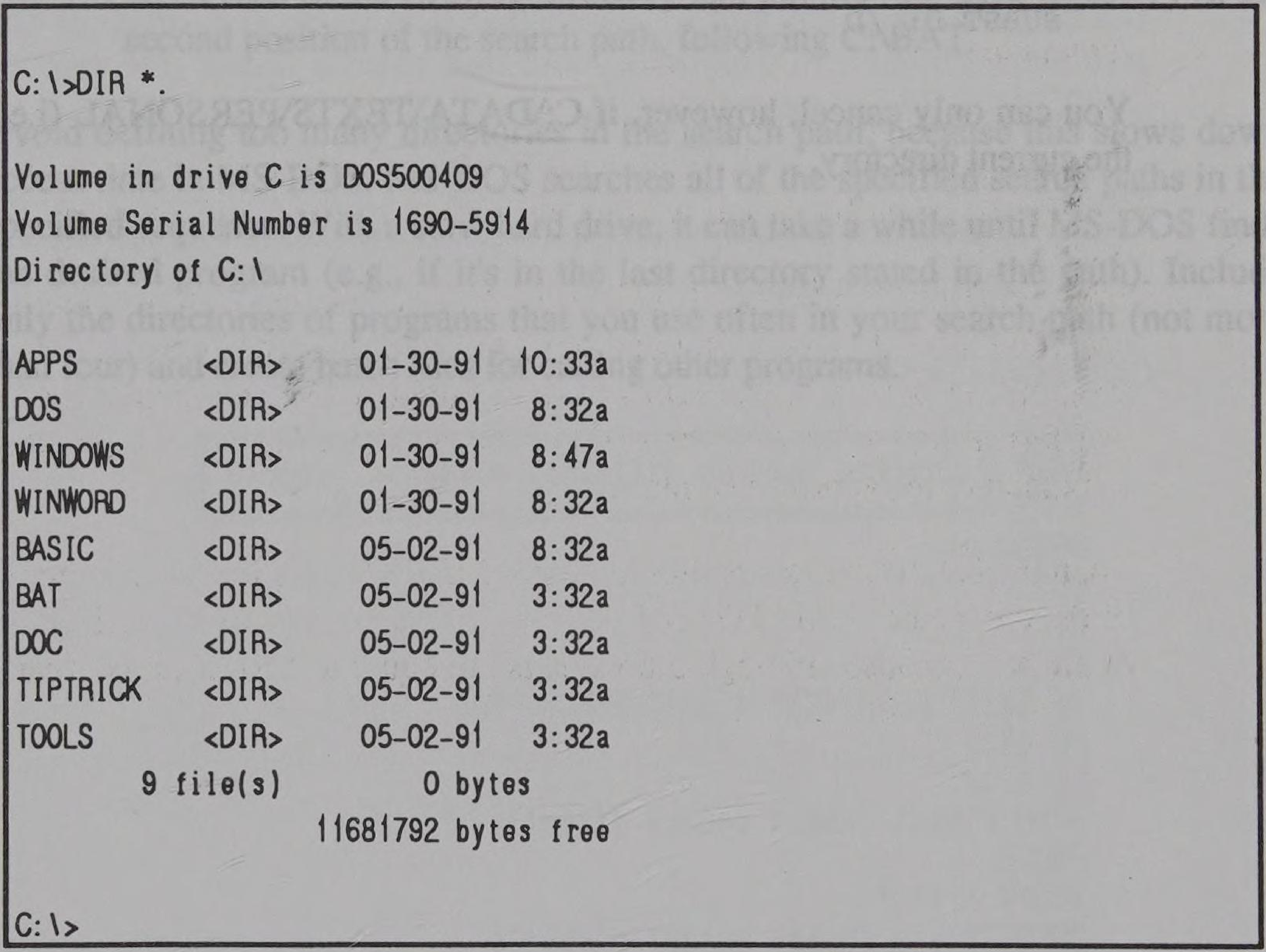
Then the batch checks whether the ERRORLEVEL is greater than the ASCII value of the character. If it didn't do this, you could also enter an "o" (ASCII 111) instead of an "n", because although the value is not equal to 110 (n), it is greater.

17.32 Displaying Subdirectories

Sometimes users have a directory with some subdirectories and files on the hard drive and want to find a certain subdirectory. It's difficult to find the subdirectory in the display of so much data. Simply type the following command:

```
DIR *.
```

This displays only the entries that don't have extensions, usually subdirectories. If you gave an extension to any of the subdirectories, they won't be displayed, while all of the files that don't have extensions will also be displayed with the subdirectories.



```
C:\>DIR *.

Volume in drive C is DOS500409
Volume Serial Number is 1690-5914
Directory of C:\

APPS         <DIR>      01-30-91  10:33a
DOS          <DIR>      01-30-91   8:32a
WINDOWS      <DIR>      01-30-91   8:47a
WINWORD      <DIR>      01-30-91   8:32a
BASIC        <DIR>      05-02-91   8:32a
BAT          <DIR>      05-02-91   3:32a
DOC          <DIR>      05-02-91   3:32a
TIPTRICK     <DIR>      05-02-91   3:32a
TOOLS        <DIR>      05-02-91   3:32a
        9 file(s)          0 bytes
                    11681792 bytes free

C:\>
```

ASCII.TXT output

17.33 Simplifying Access To A Directory

In many cases users can save themselves the trouble of specifying a path by defining directories as pseudo disk drives. For example, if you don't have disk drive D: and you frequently work with the directory, C:\DATA\TEXTS\PERSONAL, then you can make this directory a pseudo disk drive, making it a lot easier to use:

```
SUBST D: C:\DATA\TEXTS\PERSONAL
```

Then, all you have to do is enter D: when you want to use C:\DATA\TEXTS\PERSONAL. You can cancel this substitution by typing the following command line:

```
SUBST D: /D
```

You can only cancel, however, if C:\DATA\TEXTS\PERSONAL (i.e. D:) is not the current directory.

17.34 Directory Sequence And Search Path

You should specify the MS-DOS directory in the search path first, since MS-DOS searches for commands in the order in which the directories appear in PATH. By specifying the MS-DOS directory first, these commands can be found more quickly. For example, the following is the preferred path specification:

```
PATH C:\DOS;C:\BAT;C:\WORD
```

One exception to this rule is when you want to replace MS-DOS commands with batch files of the same name. Then you should put the batch files in the C:\BAT directory and put the MS-DOS directory in the second position of the search path, following C:\BAT.

Avoid defining too many directories in the search path, because this slows down access time in MS-DOS. MS-DOS searches all of the specified search paths in the specified sequence. With a slow hard drive, it can take a while until MS-DOS finds the desired program (e.g., if it's in the last directory stated in the path). Include only the directories of programs that you use often in your search path (not more than four) and create batch files for calling other programs.

17.35 Copying Different Diskette Formats

Have you ever tried to copy a 360K diskette onto a 1.2 Meg diskette?

```
DISKCOPY A: A:
```

You can't do it.

Incompatible diskette format

Since DISKCOPY copies tracks rather than files, it can't handle two different diskette formats because the two formats have differing numbers of tracks and sectors.

The solution: copy the disk by files rather than tracks (i.e., read all of the files, save them somewhere temporarily and then rewrite them). The next batch file, COPYDISK.BAT, does exactly that.

This batch file only works correctly if the source diskette doesn't contain any subdirectories.

COPYDISK.BAT

```
@REM ╔═══════════════════════════════════════════════╗
@REM ║ Copying between different diskette formats ║
@REM ╚═══════════════════════════════════════════════╝
@ECHO off
CLS
C:
CD \
IF EXIST C:\4712\*.* GOTO ERROR_TRAP
MD 4712
CD 4712
ECHO Please insert source diskette
PAUSE
XCOPY A:\*.*
ECHO Please insert target diskette
PAUSE
XCOPY *.* A:\
DEL *.*
CD ..
RD 4712
GOTO ENDITALL
:ERROR_TRAP
@ECHO Directory already exists on drive C:
:ENDITALL
@ECHO ON
```

If you have enough memory, you can use a RAM disk instead of the hard drive for temporary storage. The RAM disk can speed up the process considerably.

17.36 Multiple Versions Of MS-DOS

Switching to a new version of MS-DOS is always problematic, especially when the changes are as drastic as the ones users encounter when switching from MS-DOS 3.3 to MS-DOS 5.0. One option you have is to install both versions until you are familiar with the new version.

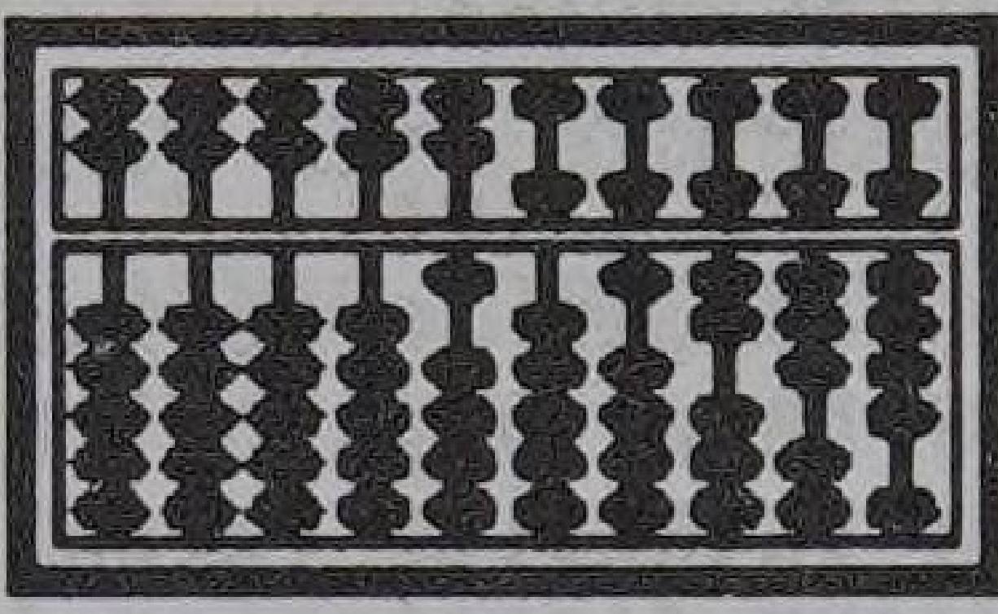
There are two methods of doing this: you can always boot the new version from a diskette, or you can install it in its own partition. If you only set up a small partition (2 to 3 Meg), you should be able to handle the "lost space". To switch back and forth between the two versions of MS-DOS, just use the FDISK program to activate the appropriate partition and reboot - you're already working with the new version of MS-DOS.

17.37 Conserving Hard Drive Space

Although MS-DOS 5.0 gives you the option of setting up partitions larger than 32 Meg, we don't recommend doing so. The bigger the partition, the bigger each cluster, and files can only occupy integral cluster numbers. In extreme cases, a file 1 byte in length could take up 2K or more on the hard drive.

Hard drive optimization programs also take a lot longer, because they have to sort through a lot more data. One alternative is to keep applications in one partition and data in another partition.

Chapter



18

18. The MS-DOS 5.0 Commands

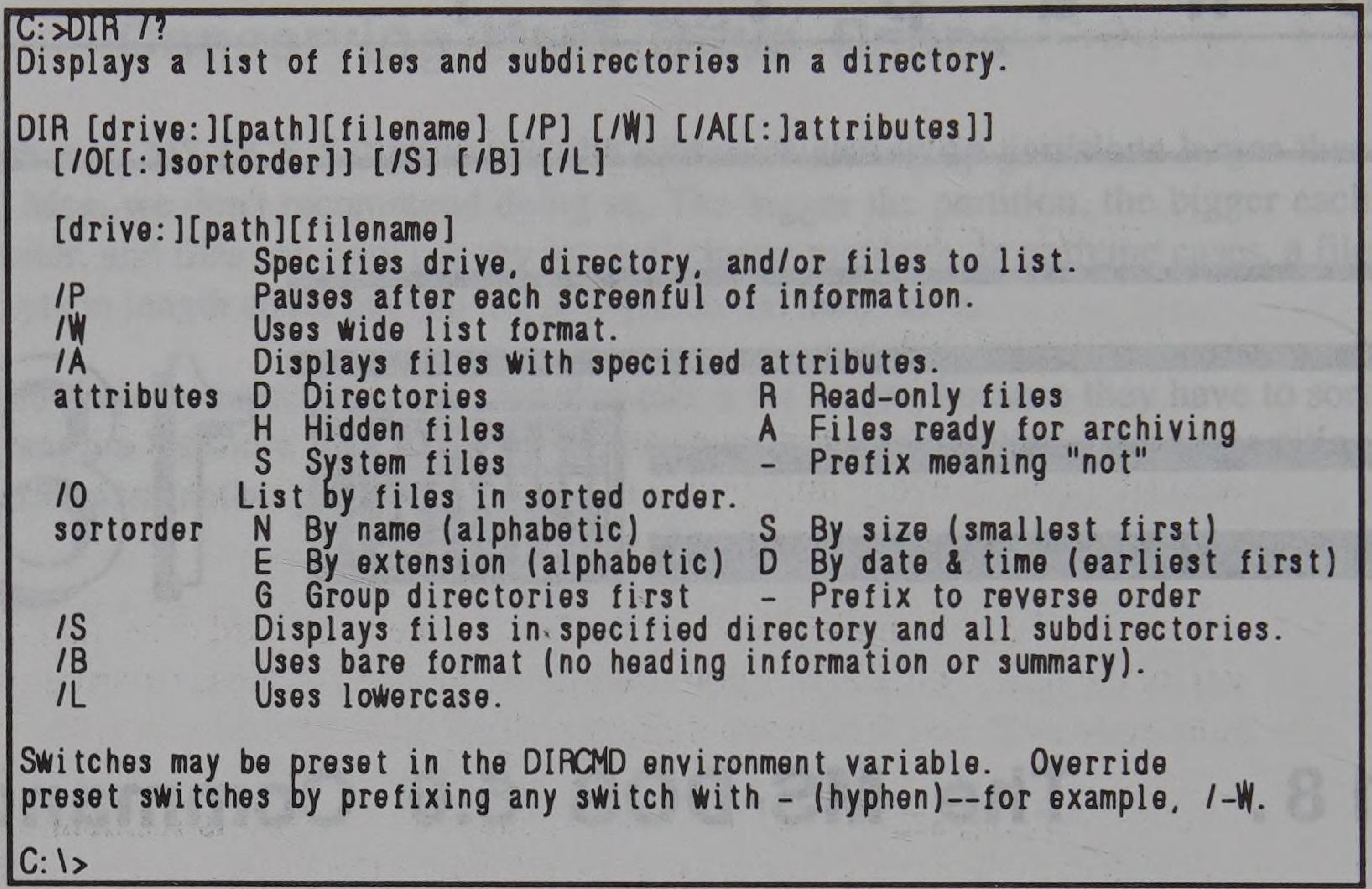
When working with MS-DOS, you often need to know the name, syntax and parameters for a specific command.

One added feature to MS-DOS 5.0 is the online help for every command. To see a list of all MS-DOS commands, type the following at the system prompt:

```
HELP
```

This command displays a list of all commands, with a brief description of the function of each command.

If more information about a specific command is needed, type the command followed by the switch "/?". This displays all of the parameters and switches for the command. The following figure shows the results of typing DIR /? at the MS-DOS command line.



```
C:>DIR /?
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/P] [/W] [/A[[:]attributes]]
  [/O[[:]sortorder]] [/S] [/B] [/L]

  [drive:][path][filename]
              Specifies drive, directory, and/or files to list.
  /P          Pauses after each screenful of information.
  /W          Uses wide list format.
  /A          Displays files with specified attributes.
  attributes  D  Directories                R  Read-only files
              H  Hidden files               A  Files ready for archiving
              S  System files               -  Prefix meaning "not"
  /O         List by files in sorted order.
  sortorder   N  By name (alphabetic)       S  By size (smallest first)
              E  By extension (alphabetic)  D  By date & time (earliest first)
              G  Group directories first    -  Prefix to reverse order
  /S          Displays files in specified directory and all subdirectories.
  /B          Uses bare format (no heading information or summary).
  /L          Uses lowercase.

Switches may be preset in the DIRCMD environment variable.  Override
preset switches by prefixing any switch with - (hyphen)--for example, /-W.

C:\>
```

Entering DIR /? displays information about using the DIR command

However, sometimes you may need additional information. This chapter provides information about all the MS-DOS commands.

18.1 Types Of MS-DOS Commands

MS-DOS commands can be divided into two groups:

- resident commands
- external commands

A *resident* command can be used at any time, because it is copied into a protected area of memory every time you start your computer. You can execute a resident command simply by typing its name.

An *external* command is stored on the DOS diskette or hard drive and has the file extension .COM or .EXE. If you want to execute one of these external commands, it must be stored on the current diskette or hard drive.

In the following section, we indicate whether a command is resident or external. Either term will appear in parentheses following the DOS command name (e.g., FIND (external).

All MS-DOS commands for Version 5.0 can be found in this chapter.

18.2 Command Reference

In this section we'll present the MS-DOS commands listed alphabetically.

The commands are described in Syntax, Switches and Description sections. The Syntax section for each command provides all of the possible forms of that command, the Switches section lists the switches that can be used with the command and the Description section explains each of the commands in more detail.

The following is an explanation of the notation used in these sections:

[ ]	Parameters enclosed in square brackets are optional. Parameter names are indicated by italic type, such as *filename*. This helps you distinguish parameters and commands from one another.
|	Alternative parameters are separated by a vertical bar. Only one of the parameters separated by a vertical bar can be entered in the command.
path	Specifies the hierarchical order of the directories separated by a backslash.
filename	Specifies the name of the file and may include an extension.
pathname	Specifies the *path* followed by the *filename*.
*	Any character or combination of characters may be replaced by an asterisk (*). This wildcard character can be located in any position of the filename or extension, but wildcards are permitted only when explicitly noted.
?	Any single character may be replaced by a question mark (?). This wildcard character can be located in any position of the filename or extension, but wildcards are permitted only when explicitly noted.

So that you can easily find them, switches are listed after each command's syntax description.

APPEND

(external)

Permits data files to be opened in specified directories as if they are in the current directory.

Syntax:	APPEND *drive: path*[;... *drive:path*] APPEND; APPEND [/e] /x[:on | :off] [/path :on | off]
path	The path in which you want to search for files. Multiple paths can be defined, separated by semicolons.
;	The semicolon, when used by itself, cancels the list of appended directories.

Switches:

/e	The search path set in APPEND is assigned to an environment variable named APPEND. The list of appended directories can be displayed with SET. This switch can only be used the first time APPEND is used after starting your system.
/x [:on | :off]	Determines whether MS-DOS is to search the appended directories when executing programs. /x:on searches appended directories. /x:off does not search appended directories. x:on must be specified the first time APPEND is used.
/path:on | path:off	Determines whether a program is to search the appended directories for a data file when the path is included with the program name. The default setting is path:on.

Description:

Sets the search pattern for data files. APPEND works like PATH, but APPEND uses the specified path when opening a data file, not a command or program file.

Normally when a file is opened, the current directory is searched for the file. When you use APPEND, then all paths listed in the APPEND search path are searched for this file. An error message is displayed if the file is not found.

This is useful, for example, when you use programs such as WordStar 3.3, which doesn't support paths. The example below checks for the file DEMO.TXT first in the DOS subdirectory then in the LIB subdirectory.

```
APPEND C:\DOS;\LIB
TYPE demo.txt
```

If all parameters are omitted, the current APPEND path is displayed. If the parameter is a semicolon, the APPEND search path is cancelled. To define multiple paths, separate each path with a semicolon. Blank spaces within the specification are not allowed.

APPEND is an external program. The first use of APPEND can include the /e switch, which stores the search path in an environment variable. You can then display the APPEND search path with SET. Applications can access a file through this search path. If you omit the /e switch on the first use of APPEND, then the search path is not stored in the environment area.

You can also specify /x on the first use of APPEND. This specifies that the defined paths should be searched for executable program files (extensions .BAT, .EXE and .COM). For example:

```
APPEND /e /x
APPEND c:\dos;d:\utility
```

has the same meaning as:

```
APPEND /e
APPEND c:\dos;d:\utility
PATH  c:\dos;d:\utility
```

Prior to DOS 4.0 the /x switch was used to extend the search path. First the current directory is searched and then the appended paths was searched. You should be aware of the differences between versions.

When a file is to be opened, it is searched for first in the current directory and then in the directories defined by APPEND. In some cases, you may not want the search to include all of the directories. Use the APPEND search path when the corresponding filename is defined without a drive statement and without a directory name. So if the following filename is specified:

```
DEMO.TXT
```

the entire APPEND search path is used. If a filename is specified as:

```
D:\DOS\KEYBOARD.SYS
```

then the search path from APPEND is not used because an explicit filename was specified. APPEND searches for a file in the appended directories with the /path:on switch. If the APPEND search directory should not be used for files that have a drive statement and/or a directory name, then use the /path:off switch:

```
APPEND /path:off
```

☞ When a file within the defined search path is found and is opened for reading, some applications may open a NEW file in the current directory; the original file remains unchanged.

Type:

`APPEND ;`	to clear the appended directory list
`APPEND`	to display the appended directory list

See also: PATH, SET

ASSIGN

(resident)

Redirects disk operations from one drive to a different drive.

Syntax: ASSIGN [*drive1*:[=]*drive2*: [...]]
ASSIGN /status

drive1:	The drive letter to reassign.
drive2:	The new drive to which input and output will be redirected.
/status	Displays current drive assignments.

Description:

ASSIGN redirects all disk operations from *drive1:* to *drive2:*. The equal sign (=) can be replaced by a space. For example, you can use the ASSIGN command if a program does not permit the active drive to be changed.

Multiple assignments can be combined in a single ASSIGN command:

```
ASSIGN a=c b=a
```

If *drive2*: is omitted, redirection from an earlier ASSIGN command is terminated. If no parameters are specified, all redirection is terminated.

☞ ASSIGN should be used carefully, since it "hides" the drive characteristics that a program might need for correct execution. The FORMAT, DISKCOPY and DISKCOMP commands ignore the ASSIGN command.

☞ A complete path must be specified if a file is to be opened for writing.

See also: JOIN, SUBST

ATTRIB

(external)

Displays or changes file attributes.

Syntax: ATTRIB [[+R | -R]] [[+A | -A] [+S | -S] [+H | -H]] [*drive:*] *path filename* [/s]

drive: path filename

The complete path and name of the file whose attributes are displayed or changed. Wildcards are allowed, but if a file has system or hidden attributes set, they must be cleared before any other attributes can be changed.

Switches:

+R	Set read-only flag
-R	Clear read-only flag
+A	Set the archive flag
-A	Clear the archive flag
+S	Set the system flag
-S	Clear the system flag
+H	Set file as hidden
-H	Clear hidden file attribute
/s	Includes all the entries in the current directory and any subdirectories

Description:

ATTRIB displays or changes the file attribute of a specified file or files.

The read-only flag is bit 0 of the FCB at address 00Bh. The read-only flag determines whether a file may be changed or deleted:

```
+R sets the read-only flag (file cannot be changed or deleted).
-R clears the read-only flag (file can be changed or deleted).
```

For example, to protect all of the commands in the \DOS subdirectory from being overwritten or deleted, use the following command:

```
ATTRIB +R C:\DOS\*.COM
DEL COMMAND.COM
Access denied
```

The archive flag is used by the BACKUP and XCOPY commands:

```
+A sets the archive flag
-A clears the archive flag
```

To display the status of the files, omit all of the ATTRIB command's parameters. Filenames preceded by an "R" are read-only files. Filenames preceded by an "A" have been modified or created since the last BACKUP or XCOPY:

```
ATTRIB D:\DOS\*.COM
R A    C:\DOS\DISKCOPY.COM
A      C:\DOS\LABEL.COM
R      C:\DOS\COMMAND.COM
```

DOS has an option for displaying all pertinent entries as well as the subdirectories. The following example displays all the BAK files on the hard drive C:

```
ATTRIB C:\*.BAK /s
```

See also: BACKUP, XCOPY

BACKUP

(external)

Makes a backup copy of one or more files from one storage device to another.

Syntax: BACKUP *source_drive*:[*pathname*] *target_drive*:
[/s][/m][/a][/f[:*size*]][/d:*date*][/t:*time*][/l[:[*drive*:][*path*]*logfile*]]

source_drive:pathname

The name of the drive and the path to be backed up. If *pathname* is omitted, all files on *source_drive* are copied. Wildcards are allowed.

target_drive:

The target diskette to which the files are copied. A path may not be specified. If *target_drive* is a hard drive, the files are copied into a subdirectory named \BACKUP.

Switches:

/s — Backup all files from every subdirectory of the source drive.

/m — Copy only the files which were modified since the last backup and clears the archive bit for each.

/a — Add backup files to an existing backup diskette. (Normally the target diskette is erased before BACKUP begins.)

/f[:*size*]

Specifies the size of the target disk for formatting. This switch causes BACKUP to format floppy diskettes that do not match the default size of the drive. *size* specifies the size, in kilobytes, of the diskette to be formatted. If *size* is not specified, the /f switch uses the default size of the drive.

/d:*date*

Backup only files which were modified after the specified date. The date is specified in the format as set by the country command. (In the United States, the format is MM-DD-YY.)

/t:*time*

Backup only the files which were modified after a specified time. The time is specified in the format set by the country command. (In the United States the format is HH:MM:SS.)

/l[:[*drive*][*path*]*logfile*]

Creates a log file called BACKUP.LOG on the target diskette. This file contains information about the number of disks required, date and time of backup, etc. A filename can also be specified.

Description:

BACKUP makes archive copies of files from the hard drive. If for some reason the original files are damaged or completely destroyed, the backup files can be recovered with the RESTORE command.

BACKUP's advantage over COPY is that it preserves the structure of the original directory hierarchy. The subdirectories are automatically recreated by RESTORE as needed. Also, RESTORE prompts the user to insert the correct disks if necessary. Copies created with BACKUP can be restored only with RESTORE.

DOS informs you if more than one diskette is required for the backup copy. RESTORE requires the diskettes to be in the same order that BACKUP created them. RESTORE will reject incorrect diskettes. Number the diskettes so that you know their correct order.

The /f switch permits specifying the size for formatting diskettes. Sizes may be specified according to the following table:

Value	Kbytes	Type
160	160K	Single-sided, double density, 5.25-inch diskette
180	180K	Single-sided, double density, 5.25-inch diskette
320	320K	Double-sided, double density, 5.25-inch diskette
360	360K	Double-sided, double density, 5.25-inch diskette
720	720K	Double-sided, double density, 3.5-inch diskette
1200	1200K	Double-sided, double density, 5.25-inch diskette
1440	1440K	Double-sided, double density, 3.5-inch diskette
2880	2880K	Double-sided, double density, 3.5-inch diskette

The following BACKUP command creates a specific type of backup on drive B:. This backup consists of all the files in the CPROGS directory of the current drive, which have .C filename extensions. The /m switch instructs BACKUP to include files that have been modified since the last time the user ran BACKUP.

```
BACKUP \CPROGS\*.C B: /m
```

When the copy is completed, BACKUP returns the following values to a batch file that may be tested with IF ERRORLEVEL:

Code	Meaning
0	No error during BACKUP
1	No files were found to backup
2	Some files could not be copied due to time-sharing conflicts
3	User terminated procedure by pressing Ctrl + C
4	Process terminated due to an error

See also: COPY, REPLACE, RESTORE, XCOPY

BREAK

(resident)

Sets or clears extended Ctrl + C checking.

Syntax: BREAK [on | off]

Description:

MS-DOS normally tests to see if Ctrl + C was pressed during input or output to one of the standard I/O devices. Extended checking tests for Ctrl + C during disk operations.

If the BREAK command is specified without parameters, the current status is displayed. The default status is OFF, unless previously changed by CONFIG.SYS or AUTOEXEC.BAT.

When using compilers, it's often useful to enable BREAK so that you can halt the compiler if an error is encountered:

```
A>BREAK ON

BREAK is on
```

CALL

(resident)

Calls one batch program from within another batch program.

Syntax:	CALL [*drive:*][*path*]*batchfile* [*parameter*]
batchfile	The name of the batch file to be called. *drive:* and *path* must be specified if *batchfile* is not in a defined search path.
parameter	A list of parameters used by the batch file. This list of parameters can be any length.

Description:

The CALL command lets you call a batch file from within another batch file as you would call a subroutine. After executing the batch file, the CALL command returns to the point immediately after the CALL command in the original batch file and executes the remaining commands.

You can pass parameters to the new batch file by listing them after *batchfile*.

The following example calls the batch procedure DEMO2.BAT and uses parameters of 11, 22 and 33. After DEMO2.BAT is completed, control returns to the main batch routine.

```
echo off
echo first batch
call demo2 11 22 33
echo back to first batch
```

CD (*resident*)

Displays the name of or changes the current directory.

Syntax:

```
CD [[drive: ][path]]
CD
```

drive:	The drive where the new current directory is located. If *drive*: is omitted, the current drive is assumed.
path	The name of the new current directory. The directory must exist.

Description:

CHDIR or CD displays or changes the current directory.

To display the current directory, omit the parameters.

By changing the current directory, all subsequent references to files are assumed to be found in that directory.

The directory that is hierarchically above the current directory can be abbreviated to two periods (..):

```
C:\CPROG\UTILITY>CD ..
```

is the same as typing:

```
C:\CPROG\UTILITY>CD \CPROG
```

If the current directory is \CPROG, you can select the subordinate directory, UTILITY, with the following command:

```
CD UTILITY
```

From any other directory you must specify the entire pathname:

```
CD\CPROG\UTILITY
```

See also: MD, RD

CHCP

(resident)

Displays or sets the active code page number.

Syntax: CHCP [*code_page_number*]

code_page_number
Specifies the system code pages defined by the COUNTRY command in the CONFIG.SYS file.

Description:

CHCP defines the active code page to be used for all devices that support code page switching. This code page is defined by the country command of the CONFIG.SYS file.

Before you can use CHCP, both the location of the COUNTRY.SYS file must be specified and the NLSFUNC program must be loaded into memory.

The *code_page_number* codes are:

437	U.S.
850	Multilingual
852	Slavic
860	Portuguese
863	Canadian-French
865	Nordic

See also: KEYB, MODE, NLSFUNC

CHDIR *(resident)*

Displays the name of or changes the current directory.

Syntax: CHDIR [[*drive:*][*path*]]
or
CD [[*drive:*][*path*]]

drive:	The drive where the new current directory is located. If *drive*: is omitted, the current drive is assumed.
path	The name of the new current directory. The directory must exist.

Description:

CHDIR or CD displays or changes the current directory.

To display the current directory, omit the parameters.

By changing the current directory, all subsequent references to files are assumed to be found in that directory.

The directory that is hierarchically above the current directory can be abbreviated to two periods (..):

```
C:\CPROG\UTILITY>CD  ..
```

is the same as typing:

```
C:\CPROG\UTILITY>CD \CPROG
```

If the current directory is \CPROG, you can select the subordinate directory, UTILITY, with the following command:

```
CD UTILITY
```

From any other directory you must specify the entire pathname:

```
CD\CPROG\UTILITY
```

See also: MD, RD

CHKDSK

(external)

Checks storage media for errors and displays a status report.

Syntax: CHKDSK [*drive:*]*pathname*[/f][/v]

drive:	The drive to be checked. If *drive* is omitted, the current drive is assumed.
pathname	The name of the file or files to be checked for fragmentation. Wildcards are allowed.

Switches:

/f	Correct any errors found (if possible). If this switch is omitted, errors are displayed and MS-DOS asks you if the error is corrected.
/v	Display the names of all files that were checked. This switch helps localize the errors.

Description:

CHKDSK checks the directories, files and memory allocation of a disk and displays a status report.

To check the entire disk, omit the drive and pathname parameters.

The status report contains information about the disk's total storage, remaining capacity, number of hidden files, number of directories, etc. For example:

```
Volume PROGRAMS    created 04-04-1991 10:35a
21309440 bytes total disk space
  143360 bytes in 3 hidden files
   51200 bytes in 16 directories
11530240 bytes 985 user files
 9560064 bytes available on disk

  655360 bytes total memory
  583280 bytes free
```

If CHKDSK encounters an error, an error message is displayed. For example, a file may not be linked properly, or a sector may be damaged. If /f is specified, errors are corrected when possible.

☞ You should not use CHKDSK to check drives that were referenced by the ASSIGN, JOIN or SUBST commands. The combination of drive information can lead to false error messages. Correcting these errors may destroy important data on the disk.

See also: DISKCOMP, COMP

CLS

(resident)

Clears the screen.

Syntax: CLS

Description:

CLS clears the screen, except for the system prompt and cursor.

The cursor is repositioned to the upper-left corner of the screen. The screen's attributes are not changed.

COMMAND

(external)

Starts a new command interpreter.

Syntax: COMMAND [[*drive:*]*path*][*device*] [/e:*nnnnn*][/P][/c *command*][/msg]

drive: path
The drive and directory from which COMMAND.COM is loaded. This value is stored in COMSPEC environment variable.

device
Specifies a different device to be used for command input and output (see CTTY).

Switches: /e:*nnnnn*
Specifies the size of the environment, where *nnnnn* is the size in bytes. This number is rounded up to a multiple of 16 bytes and must be in the range of 160 through 32768 bytes. The default value is 256 bytes.

/p
Installs the new command interpreter as permanent. In other words, you can't enter EXIT to end it.

/c *command*
This is a DOS command that is executed after loading the command processor which then stops.

/msg
This switch causes all error messages to be stored in memory. /P must also be specified with this switch.

Description:

The command processor COMMAND.COM is automatically loaded when DOS is started. It is responsible for interpreting user commands and batch files, displaying screen prompts, executing application programs, etc. COMMAND copies the command processor and allows you to change the environment.

It is possible to load a second command processor. However, you may not want to change the environment memory during a program's execution. In this case run the program in a copy of the command processor. This has the advantage that the environment memory of the original command processor is undisturbed. Each copy of the command processor needs about 3K.

The *environment* is a group of values that can be changed with the SET command. If a new command processor is started, the environment of the parent command processor is copied. Changes in the environment of the new command processor have no effect on the parent command processor.

If a new command processor is loaded, the environment of the current command processor is copied into the environment memory of the new command processor. Changes in the environment of the new command processor have no effect on that of the first command processor. The new command processor is terminated by the

EXIT command and the original environment is restored. The EXIT command has no effect if the command processor was entered as permanent.

/p makes the new command processor permanent. The values specifying the drive and path of the new COMMAND.COM are stored in the environment in a variable called COMSPEC. If the command processor ever becomes defective—for example, if an application overwrites part of the command processor—MS-DOS will try to reload the command processor from the location specified at COMSPEC.

/c causes the new command processor to be temporary. It executes only the command specified by *command* and terminates. Control is returned to the parent command processor. The /c switch must be the first switch specified.

Example

In the following example, the CHKDSK command is executed in a copy of the command processor.

```
C>COMMAND /c CHKDSK A:
```

The permanent command processor should be replaced with the COMMAND.COM file stored in the \DOS directory. The environment memory is defined as 512 bytes:

```
C>COMMAND C:\DOS\ /p /e:512
```

The following line specifies that DOS system messages should be stored in memory and the new command processor is made permanent:

```
C>COMMAND /p/msg
```

The following example calls a second batch file, DEMO2, inside a batch file. Give this batch the first three parameters of the main batch:

```
@echo off
cls
command /c demo2 %1 %2 %3
```

See also: SHELL (CONFIG.SYS)

COMP

(external)

Compares the contents of two files or sets of files.

Syntax: COMP [*drive:*][*pathname1*][*drive:*][*pathname2*] [/d][/a][/l][/n=*number*][/c]

drive: pathname1
The complete path and name of the first file to be compared. Wildcards are allowed.

drive: pathname2
The name of the second file to be compared. The two files must be the same length. Wildcards are allowed.

Switches:	/d	Displays difference in decimal format. The default setting is to display in hexadecimal format.
	/a	Displays differences as ASCII characters.
	/l	Displays line numbers of differences.
	/n=*number*	Compares only the specified number of lines of both files. This is useful when files are different sizes.
	/c	Ignore case when comparing ASCII characters in files.

Description:

COMP compares the contents of two files. The files may contain ASCII, binary or program data.

Any differences between the two files are displayed showing the position and contents of the mismatched bytes in hexadecimal format:

```
Compare error at OFFSET xxxx
File 1 = yy
File 2 = zz
```

If one of the files is shorter than the other, a message indicating this is displayed.

See also: DISKCOMP

COPY

(resident)

Copies one or more files from one location (directory or disk) to another. The COPY command has several different forms. This form of COPY copies individual files.

Syntax: COPY [*drive1:*][*pathname1*] [*drive2:*][*pathname2*] [/a][/b][/v]

drive1: pathname1

The source file to be copied; it is mandatory. Wildcards are allowed. If *drive1*: is omitted, the current drive is assumed.

drive2: pathname2

The target file. Wildcards are allowed. If *drive2:* is omitted, the current drive is assumed. If the filename is omitted from *pathname2*, the copy is given the same name as the source file.

Switches:	/a	Source file is in ASCII format and is terminated by Ctrl + Z.
	/b	Source file is in binary format.
	/v	Verify target file after the copy operation.

Description:

The following example copies the file ORDERS.DAT to ORDERS.BAK in the current directory:

```
A> COPY ORDERS.DAT ORDERS.BAK
```

To copy all of the files from the current directory with a file extension of .TXT to the \DATA subdirectory and rename them with "T" as the first character and the file extension .TEX, type:

```
C>COPY *.TXT \DATA\T*.TEX
```

COPY

(resident)

This form of COPY combines multiple source files into a single target file.

Syntax: COPY [*drive:*][*path*]*filename1* [+[[*drive:*] [*path*]*filename2*]...] [*drive2:*][*path*][*target*][/a][/b][/v]

drive: path filename1

The source file to be combined; it is mandatory. Wildcards are allowed. Multiple filenames are specified by separating their names with a plus sign (+).

drive2: path target

The target file resulting from combining the source files. If you omit *target*, the source files are combined into *filename1*.

Switches		
	/a	Source file(s) is in ASCII format and is terminated with Ctrl + Z.
	/b	Source file(s) is in binary format.
	/v	Verify target file after the copy operation.

The /b switch copies the entire source file(s) (including all Ctrl + Z characters). Ctrl + Z is not appended to the target file. If several source files are specified, all of the files following the /b switch are treated as binary files until explicitly disabled with the /a switch. This procedure can be used to create ASCII files from binary files or binary files from ASCII files.

The /v switch checks the target file to verify that it's the same as the original file.

If all switches are omitted or the /a switch is used, COPY copies ASCII text files. These files are terminated with a Ctrl + Z (ASCII value 26). COPY terminates the copy process after a Ctrl + Z in the source file is encountered. COPY also appends a Ctrl + Z to the end of the target file. This is not very helpful if you want to copy binary files.

COPY *(resident)*

This form of COPY copies files from *from_device* to *to_device*.

Syntax: COPY *from_device to_device*

from_device	The device or file whose data is to be copied.
to_device	The device or file to accept the data to be copied.

The following device names are allowed:

- AUX
- COM1
- COM2
- CON
- LPT1
- LPT2
- LPT3
- NUL
- PRN

Description:

You can also use COPY to send data from one device to the other. For example, you can print a file with COPY:

```
COPY READ.ME PRN:
COPY CON AUTOEXEC.BAT
```

One way COPY can be used is to create small files, such as a batch file, from the keyboard. By doing this you don't have to use a word processor or EDLIN to enter a few lines of text. To create an AUTOEXEC.BAT file, type:

```
COPY CON AUTOEXEC.BAT
```

Now you can enter text from the keyboard on separate lines. To end the file, press either Ctrl + Z or F6. Each line entered is ended by pressing Enter. These lines are stored in the file AUTOEXEC.BAT. Once you have completed a line by pressing Enter, you cannot revise or correct it.

You can also send characters to a printer directly from the keyboard. Type the following (terminate with F6 or Ctrl + Z):

```
COPY CON PRN
```

See also: BACKUP, DISKCOPY, REPLACE, RESTORE, XCOPY

CTTY

(external)

Changes the terminal device used to control your system.

Syntax: CTTY *device*

device The device that becomes the new standard input/output device.

Description:

CTTY specifies the input and output devices used. The default input and output devices are the keyboard and screen, which collectively are called the console or CON.

The following I/O devices are allowed:

Name	Device
CON	Screen and keyboard (normal)
AUX	Auxiliary port; RS-232
COM1	Communications port 1; RS-232
COM2	Communications port 2; RS-232

For example, you can convert your PC into a "dumb terminal". Assuming you have the proper hardware and have made the necessary connections, you could use the following configuration with a modem to communicate with a mainframe computer:

```
CTTY AUX
```

If you change the standard I/O device, a device capable of input and output must be connected to the AUX interface. For example, a printer would not be suitable for this because it is only capable of output, not input. To restore input and output to the standard keyboard and screen (console), type the following (from the new standard input device):

```
CTTY CON
```

To use a device other than AUX, COM1, COM2 or CON, the device must have a device driver specified by the DEVICE command in the CONFIG.SYS file.

☞ Only applications that use the DOS interrupts can reference the input and output devices defined with CTTY.

See also: MODE

DATE

(resident)

Displays or sets the system date.

Syntax: DATE[*date*]

date This is the new system date.

Description:

If *date* is omitted, the current system date is displayed and you are asked to enter a new date. The current system date is unchanged if you press Enter without entering a new date.

date has one of the following formats, depending on the country code you enter in COUNTRY of CONFIG.SYS:

```
MM-DD-YY      or
MM-DD-YYYY    or
DD-MM-YY      or
YY-MM-DD
```

MM represents the month, DD represents the day and YY or YYYY represents the year. The year can be entered as either two or four digits. The period (.), dash (-) or slash (/) can be used as the separator between the elements of the date. Which separator is used when the date is displayed depends on the setting of COUNTRY in the CONFIG.SYS file. Dashes are used in the U.S.

See also: TIME

DEBUG

(external)

Starts DEBUG, a program testing and editing tool.

Syntax: DEBUG [*drive:*][*path*]*filename* [*Switches* | *parameters*]

drive: path filename

The program or file to be analyzed with DEBUG. The file is loaded at its defined starting address. If omitted, the current contents of memory are used.

Switches: These are additional parameters. You may specify these switches after DEBUG is started.

parameters

Any command line information required by the executable file to be tested.

Description:

The DEBUG program analyzes machine language programs during execution. Memory locations can be displayed, printed, changed, stored, etc. The prompt symbol for DEBUG is the minus sign (-).

If you omit a filename when you start DEBUG, the current memory area is assumed. You can load a file with the DEBUG commands N or L.

Once DEBUG is started, the segment registers are all set to the lowest memory boundary. The flags are cleared. The instruction pointer IP is set to 0100h, which is where most programs start (at least the COM programs). All registers except for SP are set to zero. Register pair BX/CX contains the length of the program.

Addresses can be specified with or without segment specification. When the segment address is omitted, DS is the default (except for the commands G, L, T, U and W, where the code segment is assumed).

The segment registers themselves can be used as the segment address. The following address specifications are allowed:

```
400
CS:1022
SS:900
```

To specify a range of memory, use either two hexadecimal values separated by a space or a starting value followed by a length (also separated by a space). The length (in hexadecimal) is prefixed with an L. In this case, the segment is specified only once. If CS contains the value 196a, then the following six commands all specify the memory area 196A:0100 to 196A:019F:

```
100 19f
100 L1a0
CS:100 19F
CS:100 L1A0
196a:100 19F
196a:100 L1A0
```

To define a list of data, separate the hexadecimal values from one another with spaces:

```
E DS:3000 20 30 31 32 33 34 35 36 37
```

ASCII strings are enclosed by quotation marks. If a quotation mark is present in the string itself, enter it as a double quotation mark ("").

```
31 32 "345""6789"   yields       12345"6789
```

In addition to executable programs, you can also load and edit text files with DEBUG.

☞ Be careful with EXE files. These files are not saved by DEBUG because they are relocatable.

The following information provides only a brief overview of the DEBUG commands.

? Displays a list of the Debug commands.

a[*address*]

Assembles 8086/8087/8088 mnemonics. DEBUG supports the 8087 instruction set. The length of each operation, byte or word must be specified. All numeric values must be entered in hexadecimal.

```
A 1000
1af0:1000 inc byte [1fff]
1af0:1004 dec word [1ffd]
1af0:1008 db "Here is a string"
1af0:1018 db fd, 3, 09
1af0:1018 dw 2000, 9dfe
1af0:101F movsb
```

c[*range*] [*address*]

Compare two areas of memory. The first parameter, *range*, specifies the starting and ending addresses and the second parameter, *address*, is the starting address of the second memory area to compare. The length of *range* determines the length of the comparison.

If the *range* and *address* memory areas are identical, Debug displays nothing and returns to the Debug prompt. If there are differences, they are displayed in the following format:

```
address1 byte1 byte2 address2
```

address1 is in *range*. *address2* is in the second area. The two bytes are displayed side by side.

d *range*

Display (dump) the specified memory range in hexadecimal format. Both the hexadecimal and ASCII values are displayed. ASCII values which cannot be displayed are represented as periods. Sixteen bytes are displayed per line.

If only a starting address is specified, 128 bytes are displayed. If the starting address is omitted, the area at CS:0100 or the last address dumped is displayed:

```
d 3000:9222 L200
```

e *address list*

Enter data in memory. If *list* is omitted, the memory contents beginning at *address* is displayed and you can enter values to replace the current contents. To leave memory unchanged, press the Spacebar. To move back by one memory location, press the − key. To terminate this command, press the Enter key.

The following displays the contents of memory one byte at a time from 2000 thru 3331 and allows you to change the contents:

```
e 2000:3331 43 90 "Grand total:"
e 2000:3331
2000:3331 20. 01  ff. 02  31. ff
```

f *range list*

Fill a *range* with the values in *list*. For example, to fill the screen with the letters DOS, type the following command:

```
fb000:000 Lfa0 "D" 09 "O" 09 "S" 09 20 07
```

Replace B000 with B800 if you have a color monitor.

g (go) [=*address*] [*breakpoints*]

Start a program at the address CS:=*address*. This address must be the first byte of an instruction.

You can define up to 10 breakpoints. When one of these addresses is encountered, the program is halted and control is passed back to DEBUG. The breakpoints call interrupt CCh.

If =*address* is omitted, the program is started at the address CS:IP. You can use this form of the command after a breakpoint has been encountered:

```
g=0100 0110 0212 9fd
```

`h value1[,] value2`

Adds the first two parameters *value1* and *value2*, then subtracts the second parameter from the first. *value1* and *value2* can be any hexadecimal number between 0 and FFFFh.

```
-h 2c78 421
3099 2857
```

If *value2* is larger than *value1*, the result is represented as a two's complement number.

```
-h 3d00 3e00
7B00 FF00
```

`i port`

Reads and displays one byte of data from the specified port. The value is displayed in hexadecimal format.

`l address drive sector num_sectors`

Load a file or specified disk sectors into memory.

Parameters

address

Specifies the memory location where the file or disk sector should be loaded.

drive

Specifies the drive which contains the disk from which sectors are to be read. This is a numeric value: 0=A, 1=B, 2=C, etc.

sector

Hexadecimal number of the first sector to load.

num_sectors

Hexadecimal value of the number of consecutive sectors to load.

Using L without parameters loads the file specified when DEBUG was started or the file was last defined by the N command. The filename cannot be specified from the L command. The file is loaded starting at CS:0100, or at *`address`*, if specified.

The other form of L loads a specified number of *`sectors`* from *`drive`*, starting at memory location *`address`*, beginning at the relative sector, *`sector`*.

`drive` is not a letter but a digit from 0 to 255, where 0=A, 1=B, etc.

l	load	the last file
l 4000	load	the last file at 04000h
l 200 2 0 D	load	13 sectors from drive c, starting with sector 0, to address SC:200

`m` *`range address`*

Copies the contents of memory at *`range`* to *`address`*.

`range`

The starting and ending address, or the starting address and length, of the memory to be copied to *`address`*.

`address`

The starting address of the location to which the contents of *`range`* is to be copied.

`n` *`filename`*

Sets the filename for subsequent load (l) and write (w) commands.

```
ntextomat.exe
```

`o` *`address byte`*

Output the data value *`byte`* to port *`address`*:

```
o 02f8 2e
```

`p` *`[=address][number]`*

Proceed; executes a loop, a repeated string instruction, a software interrupt or a subroutine. May also be used to trace through any other instructions.

`=address`

The location of the first instruction to execute. If no address is specified, the default address in the CS:IP is used.

`number`

Specifies the number of instructions to execute before returning control to Debug.

`q`

Quit DEBUG without confirmation and without saving the current file. If you have made changes that you want to save, you must save them before quitting DEBUG.

`r register`

Display and/or change the register contents.

R without parameters displays the contents of all registers.
R `register` displays the contents of the corresponding registers.

You can change the contents of these registers. The following register names are recognized by DEBUG:

```
AX BX CX DX CS DS ES SS DI SI BP SP F IP (or PC)
```

You can change more than one flag register by separating the register names with spaces.

The following codes can be used to change the corresponding flag:

Flag	Flag set	Flag cleared
Overflow	OV	NV
Direction	DN (Down)	UP (Up)
Interrupt	EI (Enabled)	DI (Disabled)
Sign	NG (Negative)	PL (Plus)
Zero	ZR (Zero)	NZ (Nonzero)
Aux Carry	AC (Aux Carry)	NA (No Auxiliary)
Parity	PE (Parity even)	PO (Parity odd)
Carry	CY (Carry)	NC (No Carry)

For example, the following command clears the interrupt and carry flags in the flag register:

```
-r f
OV DN EI ZR NA PO CY -   di nc
```

`s range list`

Search for the characters `list` within memory `range`. The address at which the characters are found is displayed in the form *segment:offset*. For example, if you're searching for all occurrences of the string "error", use the following command:

```
s0100 ffff "error"
6abd:3a94
```

The search is case sensitive. For example, the above example will find "error", but not "ERROR".

`t<=start number`

Trace the program in memory beginning at address `start` for `number` executions. If `start` is omitted, execution begins at the current value of CS:IP. If `number` is not specified, a single instruction is executed. When the specified number of instructions have been executed, the following is displayed:

```
t           start at CS:IP    (1 instruction)
t=0400      start at CS:0400  (1 instruction)
t=0400 8    start at CS:IP    (8 instructions)
```

`u range`

Unassemble (disassemble) memory at `range`. The hexadecimal opcodes are displayed in their equivalent mnemonic form as follows:

```
u2000
20FE:2000  BAFFFE       MOV    DX,FEFF
20FE:2003  2E           CS:
20FE:2004  8A0E2020     MOV CH,[2020]
```

If you enter only a starting address, 32 bytes of memory are disassembled. If you omit `range`, the 32 bytes of memory at CS:IP are disassembled.

`w[address] [drive] [first_sector sectors]`

Write the data to disk.

w without parameters or with `address` writes the file specified when DEBUG was started or last defined by the N command.

Using parameters writes the specified number of sectors to `drive`, starting at memory location `address`, beginning at relative sector `first_sector`.

`drive` is a hexadecimal value with 0=A, 1=B, 2=C, etc. Files with .EXE or .HEX extensions cannot be rewritten—DEBUG will display an error message if you try to do this.

Be careful when you write directly to sectors. Under certain conditions, writing directly to sectors can overwrite important data, the FAT or the directory.

`XA count`

Allocates a specified number of pages of expanded memory.

`count`

The number of 16K pages of expanded memory to allocate.

If this command executes successfully, a message, similar to the following, is displayed:

```
Handle created=0004
```

`XD` *`handle`*

De-allocates a handle to expanded memory.

Example

```
XD 0004
```

This example de-allocates handle 0004. If successful, DEBUG displays the following message:

```
Handle 0004 deallocated.
```

`XM [`*`lpage`*`] [`*`ppage`*`] [`*`handle`*`]`

Maps a logical page of expanded memory, associated with a specific handle, to a physical page of expanded memory.

`lpage`

The number of the logical page of expanded memory to map.

`ppage`

The number of the physical page to which *lpage* is mapped.

`handle`

The handle associated with a logical page of expanded memory.

`XS` Displays information about expanded memory in the following format:

Handle *x* has *x* pages allocated
Physical page *x* - Frame segment *x*
x of a total *x* EMS pages have been allocated
x of a total *x* EMS handles have been allocated

DEL

(resident)

Deletes specified files.

Syntax: DEL [*drive:*][*path*] *filename* [/p]

[*drive:*][*path*] *filename*
Specifies the drive, path and filename of the file or files to delete. Wildcards are permitted as filenames.

Switches: /p Prompts for confirmation before erasing a file.

If the /p switch is used, a prompt appears asking for confirmation before each file is deleted.

To erase all files in the DOC directory of drive C:, type:

```
DEL C:\DOC\*.*
```

or

```
ERASE C:\DOC\*.*
```

See also: ERASE, UNDELETE

DELOLDOS

(external)

Removes previous version of DOS (saved in the OLD_DOS.1 directory when MS-DOS 5.0 is installed).

Syntax: DELOLDOS

DELOLDOS removes all files saved in the OLD_DOS.1 directory from your disk drive. This command also removes itself from the DOS directory after running because there is no further use for it.

When you install MS-DOS 5.0, the previous version of DOS is saved in case you want to reinstall this version for any reason.

If MS-DOS 5.0 is installed more than one time without running DELOLDOS, multiple copies of your DOS directory will be created named OLD_DOS.1, OLD_DOS.2, etc.

DIR

(resident)

Displays a list of files and subdirectories in the current directory.

Syntax: DIR [*drive:*][*path*][*filename*][/w][/p][/a[[*attributes*]
][/o[[:]*sortorder*]][/s][/b][/l]

drive:	The drive containing the directory. If you omit *drive:*, the current drive is assumed.
path	The subdirectory to be displayed. If you omit *path*, the current path is assumed.
filename	The name of a specific file to be displayed. If you omit *filename*, the entire directory is displayed. Wildcards are allowed.

Switches:

/w	Wide format switch: Displays the filenames across each line with as many as five filenames per line. No other information about each file is displayed.
/p	Page switch: Displays the filenames one screen page at a time. You can view the next page by pressing any key.
/a[:]	Displays only the names of files and directories with specified attributes. (See the table below.)
/o[:]	Display files by specified sort order (see table below).
/s	All files and subdirectories in specified directory.
/b	Bare format (no heading information or summary).
/l	Display in lowercase.

Description:

DIR displays the directory of the files on a disk, the number of files in the directory and the number of available bytes on the disk.

To display the external COM commands for DOS (assuming you're in the correct subdirectory), use the following command:

```
DIR *.COM
```

If these external COM commands are in a different subdirectory named DOS, the following command could be used:

```
DIR \DOS\*.COM
```

The following commands display the files in the current directory of a specified drive:

```
DIR
DIR A:
```

The following commands display all of the files in the \DOS directory:

```
DIR \DOS
DIR \DOS\*.*
```

The wildcard parameter *.* (representing all of the files) can be omitted because it is the default.

To print a hardcopy of the directory, redirect output to the printer. Do this with the > symbol:

```
DIR *.COM > PRN
```

It's also possible to display only files with specific file attributes. The following table lists the attribute codes to use with the /a: switch:

Attribute Code	Meaning
h	Hidden files
-h	Files that are not hidden
s	System files
-s	Files other than system files
d	Directories
-d	Files only (no directories)
a	Files with archive bit set
-a	Files with archive bit cleared
r	Read only files
-r	No read only files

For example, to display a directory of only those files in the current directory which have their archive bits set, you could use the following command line:

```
DIR /a:a
```

To display the filenames in a different order, use the /s:order switch to specify the order as shown below:

Sort order Code	Meaning
n	Alphabetical order by name
-n	Reverse alphabetical order
e	Alphabetical order by filename extension
-e	Reverse alphabetical order by filename extension
d	By date and time, earliest first
-d	By date and time, last first
s	By size, smallest to largest
-s	By size, largest to smallest
g	Directories grouped before files
-g	Directories grouped after files

To display the filenames in the current directory sorted alphabetically by name, type:

```
DIR /o:n
```

Using the /s switch displays all files in the current directory, as well as files in any subdirectories.

/b displays the files with no heading and summary information.

/l displays an unsorted list of file and directory names in lowercase letters. This switch does not convert external characters to lowercase.

You can also preset these switches by setting the DIRCMD environment variable.

DISKCOMP *(external)*

Compares the contents of two floppy diskettes.

Syntax: DISKCOMP [*drive1*:[*drive2*:]][/1][/8]

drive1:	The drive containing the first diskette.
drive2:	The drive containing the second diskette. If *drive2:* is omitted, the current drive is assumed.

Switches:

/1	Compare only the first sides of double-sided diskettes.
/8	Compare only the first eight sectors of each track, regardless of the number of sectors per track.

Description:

DISKCOMP compares the contents of two diskettes. These diskettes are compared track by track. Any differences are displayed; the location of the mismatched data is indicated.

Remember that two disks which contain the same files are not necessarily identical. If you are only interested in comparing the contents of files, use the COMP command.

DISKCOMP works only with floppy diskettes. An error message is displayed if a hard drive is specified.

If the current drive is A: and both *drive1:* and *drive2:* are omitted, DOS assumes you want to use a single drive (A:) and asks you to insert one diskette at a time into drive A:.

☞ Don't use DISKCOMP with drives that have been processed with the ASSIGN, JOIN or SUBST commands.

See also: COMP

DISKCOPY

(external)

Copies the contents of one floppy diskette to another.

Syntax: DISKCOPY [*drive1*:][*drive2*:][/1][/v]

drive1:	The drive containing the source diskette to be copied. If omitted, the current drive is assumed.
drive2:	The drive containing the target diskette. If omitted, the current drive is assumed.

Switches:

[/1] *one*	Copies only the first side of the source diskette, even if both sides are formatted. The target diskette is also formatted as single-sided.
[/v]	Verifies that the information is copied correctly.

Description:

DISKCOPY makes a copy of the source diskette, track by track. If you have only one drive, DOS assumes that you are copying to the same drive and prompts you to insert the correct diskette as necessary.

If the target diskette is not formatted before the copy operation, DOS recognizes this and reformats the diskette with the same number of sides and sectors per track as the source diskette. If errors occur during copying, they are displayed; the side, track and sector are indicated.

If the source diskette has a volume serial number, DISKCOPY creates a new volume serial number for the destination diskette and displays this number when the copy operation is complete.

Specify the /1 switch to copy only the first side of the source diskette. If the target diskette is unformatted, it's then formatted as single-sided.

When the /v switch is used, DISKCOPY checks that all information is copied correctly. Verifying the copy operation requires additional processing time.

Since DISKCOPY makes an exact copy of the source disk, it does not eliminate any file and disk fragmentation. To de-fragment the files, use the COPY or XCOPY commands.

The ERRORLEVEL parameter of the IF command line may be used in a batch file to process exit error codes returned by DISKCOPY.

DISKCOPY Exit Error Codes

0	The copy operation was successful.
1	A nonfatal read/write error occurred.
2	The Ctrl+C key combination was pressed by the user.
3	A fatal error occurred.
4	An initialization error occurred.

☞ Don't use DISKCOPY with drives that have been processed with the ASSIGN, JOIN or SUBST commands.

See also: BACKUP, COPY, DISKCOMP, REPLACE, RESTORE, XCOPY

DOSKEY *(external)*

Allows command line editing. Recalls and edits commands and defines macros.

Syntax: DOSKEY [/reinstall][/bufsize=*size*]
[/macros][/history][/insert | overstrike]
[*macroname*=[*text*]]

/reinstall	Installs a new copy of the DOSKEY program, even if one is already installed.
/bufsize=*size*	Specifies the size of the command history buffer for storing commands and macros.
/macros	Displays a list of all DOSKEY macros.
/history	Displays the contents of the command history buffer.
/insert | overstrike	Determines how new characters entered in a command line will be handled. INSERT specifies that new text will be inserted into old text; OVERSTRIKE specifies that new text is to write over previous text.
macroname	Specifies a name for a created macro.
text	Specifies commands to be recorded.

Description:

The DOSKEY is a Terminate and Stay Resident (TSR) program that lets you automate and customize MS-DOS command lines. DOSKEY requires about 3K of memory.

The following keys have special uses from the command line when DOSKEY is installed:

↑	Recalls the MS-DOS command issued prior to the command line being displayed. Subsequent key presses move further back through the command history.
↓	Recalls the MS-DOS command issued after the command line being displayed. Subsequent key presses move further ahead through the command history.
PgUp	Recalls the oldest MS-DOS command in the history buffer.
PgDn	Recalls the most recent MS-DOS command issued.

The DOSKEY command lets you edit a displayed command line. If INSERT has been set, all keystrokes are inserted into the command line at the cursor location, while OVERSTRIKE replaces the character at the cursor position with the newly typed character.

Other editing keys and their functions are used as follows:

←	Moves the cursor back one character.
→	Moves the cursor forward one character.
Ctrl+←	Moves the cursor back one word.
Ctrl+→	Moves the cursor ahead one word.
Home	Moves the cursor to the beginning of the line.
End	Moves the cursor to the end of the line.
Esc	Clears the command line.
F1	Copies one character from the template to the command line. (The template is a buffer that contains the last command entered.)
F2	Searches forward in the template for the next key entered after pressing F2. The text up to, but not including, the character pressed is inserted on the command line.
F3	Copies the remaining characters from the template beginning at the position in the template which corresponds to the cursor position on the command line.
F4	Deletes the command line in the template up to the next key pressed.
F5	Copies the current command line into the template and clears the command line.
F6	Places an end-of-file mark (Ctrl+Z) at the end of the current command line.
F7	Displays all commands in memory.
Alt+F7	Erases all command lines stored in memory.

F8	Searches memory for a command line matching the current command line. Type the first part of the command line to search for, then press F8. DOSKEY displays the first command line matching the text entered.
F9	Asks for the number of the command line to display. (Press F7 to display the commands with numbers.)
F10	Display all macros.
Alt+F10	Clear all macros.

DOSKEY can be used to create macros. The following special characters can be used to perform command operations when defining a macro:

$G or $g	Redirects output. Sends output to a device or file instead of the default display device.
$L or $l	Redirects input. Takes input from a device or a file instead of the keyboard.
$B or $b	Pipes commands. Sends the output of the macro to another command.
$T or $t	Command separator. Separates commands when creating a macro or entering commands directly.
$$	The dollar sign filename character. This is equivalent to $.
$1 through $9	Replaceable parameters. Use $1 through $9 to specify parameters that may be entered from the command line when the macro name is entered. The $1 character is equivalent to %1 in batch files.
$*	Replaceable parameter in macros.

DOSSHELL

(external)

Starts the MS-DOS Shell.

Syntax:

```
DOSSHELL [/t[:res[n]]][/b]
DOSSHELL [/g[:res[n]]][/b]
```

Switches:

/t	Starts the MS-DOS Shell in text mode.
:res[*n*]	A letter representing the screen resolution (L=low, m=medium, h=high).
/b	Starts MS-DOS Shell using black-and-white color scheme.
/g	Starts MS-DOS Shell in graphics mode.

Description:

The MS-DOS Shell user interface is started by typing the DOSSHELL command. If your computer does not use a hard drive, the appropriate diskette must be in the drive. You do not need to define any parameters to configure the MS-DOS Shell because these are given in the DOSSHELL.BAT batch file.

If you leave the Shell by pressing Shift + F9 or with a corresponding menu command, you can restart the Shell again by typing the EXIT command; you don't need to run the DOSSHELL command again.

The DOSSHELL command may be added to your AUTOEXEC.BAT file to start the Shell whenever your computer is started.

ECHO

(resident)

Turns on or off the command message display.

Syntax: ECHO [on | off]
ECHO [*message*]

on | off	Specifies whether the command-echoing feature is on or off. To display current ECHO setting, enter ECHO with no parameters.
message	Displays specified text on the screen.

ECHO is especially useful with batch files. When ECHO is set to off, ECHO *message* can be used to display a message without displaying all other command lines.

EDIT

(external)

Starts the MS-DOS Editor for creating and changing ASCII files.

Syntax: EDIT [[,*drive:*][*path*]*filename*][/b][/g][/h][/nohi]

drive: path filename

The name of the file to be edited. If it does not exist, it is created.

Switches:

/b	Permits a monochrome monitor to be used with a color graphics card.
/g	Provides fastest screen updating for a CGA monitor.
/h	Displays the maximum number of lines possible for the monitor you are using.
/nohi	Enables using 8-color monitors with the MS-DOS Editor. Usually MS-DOS uses 16 colors.

The MS-DOS Editor is used for editing ASCII files quickly and easily. If the QBASIC.EXE file is not in the current directory or search path or not in the same directory as EDIT.COM, you cannot use the MS-DOS Editor.

The MS-DOS Editor uses predefined keys and menus to edit an ASCII file with a full screen editor. This is a vast improvement over EDLIN which was supplied with previous versions of MS-DOS.

For more information about the MS-DOS Editor, refer to Chapter 13.

EDLIN

(external)

Starts the line editor.

Syntax: EDLIN [*drive:*][*path*] *filename* [/b]

drive: path filename
The name of the file to be edited. If it does not exist, it is created. You cannot edit a file with a .BAK extension.

Switches: /b File to be loaded is a binary file. The end-of-file marker Ctrl+Z is ignored.

Description:

EDLIN is a text line editor with very simple commands. EDLIN can be used to create or edit small batch files. EDLIN loads a file until it encounters a Ctrl+Z (ASCII 26 or 01Ah). To ignore this end-of-file marker, use the /b switch. EDLIN has the following commands:

Command	Syntax
Append	*line_number* a
Copy	*first_line,last_line,target num* c
Delete	*first_line,last_line* d
Edit	*line*
End (Save file)	e
Help	?
Insert	*line* i
List	*start_line,end_line* l
Move	*first_line,last_line* m
Page display	*first_line,last_line* p
Quit (disregard changes)	q
Replace	*first_line,last_line* ?r *oldtext* ^Z *newtext*
Search	*first_line,last_line* ?s *string*
Transfer	*targett*[*drive:*][*path*]*filename*
Write	*lines* w

A text file is composed of one or more lines. Each line may contain up to 253 characters and is preceded by a line number when displayed.

The line numbers are not part of the text file, but are automatically maintained by EDLIN. After each EDLIN command that inserts or deletes a line of text, the lines are renumbered to readjust.

All EDLIN commands consist of a single letter preceded by parameters. You can use either uppercase or lowercase characters. If the line number is not specified, the last line processed is assumed. The last line processed is designated by an asterisk. Press Enter without a command to edit this line.

You can enter multiple commands on a line, separated by semicolons (;).

In addition to specifying an absolute line number, you can use the following characters to represent a position relative to the current line number or a symbol for certain line numbers:

Symbol	Meaning
#	The last line number in the EDLIN buffer
.	The current line
+ | -	A line number relative to the current line

The I command is used to enter new lines. Here is an example of creating an AUTOEXEC file:

```
EDLIN AUTOEXEC.BAT
New file
*i

     1:*echo off
     2:*cls
     3:*echo Please enter the date and time:
     4:*echo -
     5:*date
     6:*time
     7:*^z
*q
Abort edit (Y/N)? N
*e
```

To save the edited text, you must exit EDLIN with the E command. A backup copy of the file with a .BAK extension is made before the file is edited.

If you exit EDLIN with the Q command, DOS asks you to confirm this without saving the changes.

EMM386 *(external)*

Turns EMM386 expanded memory on or off.

Syntax: EMM386 [on | off][w=on | w=off]

on | off	Activates or suspends EMM386.EXE device driver or places it in auto mode. (Auto mode enables expanded memory support only when a program calls for it.) Default value is ON.
w=on | w=off	Enables or disables Weitek co-processor support. The default value is w=off.

The EMM386 device driver must have been installed using the DEVICE command from your CONFIG.SYS file before using this command. Your computer must have an 80386 (or higher) microprocessor.

ERASE

(resident)

Syntax: ERASE [*drive:*][*path*] *filename* [/p]

drive: path filename
The path and name of the file to be deleted. Wildcards are allowed.

Switches: /p Prompts for confirmation before erasing a file.

Description:

This command is identical to DELETE and uses the same syntax.

ERASE removes a file from disk. The files are not immediately physically erased, but the filename in the directory is marked as erased and the sectors occupied by the file are marked as free (and thus subject to overwriting).

If you attempt to erase all of the files on a drive or a directory, DOS asks for confirmation. If you don't confirm the operation, the command is terminated:

```
ERASE FILE.DAT
ERASE *.HLP
ERASE O??O.*
ERASE *.*
All files in directory will be deleted!
Are you sure (Y/N)?n
```

If the /p switch is used, a prompt appears asking for confirmation before each file is deleted.

To erase all files in the DOC directory of drive C:, type:

```
ERASE C:\DOC\*.*
```

See also: DEL, UNDELETE

EXE2BIN

(external)

Converts an EXE file to a BIN file.

Syntax: EXE2BIN [*drive1*:]*path1 input_file*[*drive2*:] [*pathname2 output_file*]

drive: pathname1	The complete path for the input file.
drive: pathname2	The complete path for the output file.
input_file	The name of the input file.
output_file	The name of the output file. If omitted, a BIN file with the same filename as the input is assumed.

Description:

EXE2BIN converts an executable file (EXE) into a binary file. It is used by programmers, not general PC users.

EXE files are created by an assembler or compiler. EXE files can be executed by simply entering the name of the file. In contrast to EXE files, COM files are neither relocatable nor segment-oriented. They always start at address 0100h.

There are some restrictions to remember when using EXE2BIN.

The input file must be a valid EXE file produced by the linker and must not be packed. The EXE file may not contain a stack segment. The program and data segments may not exceed 64K.

If the EXE file does not start at address 0100h and the entry point is not defined with the END pseudo-op of the assembler or if a segment-oriented command occurs, EXE2BIN will simply create a binary file. This binary file cannot be correctly loaded and executed by the COMMAND.COM command processor. In this case, EXE2BIN requests a starting address for this binary file. This binary file can be loaded and started only from an application.

EXIT

(resident)

Quits the COMMAND.COM program and returns to the program which started it, if one exists.

Syntax: EXIT

Description:

When a second command processor is loaded in addition to the permanent command processor COMMAND.COM, the alternate command processor is terminated with EXIT (see also COMMAND). If you load multiple command processors, each copy takes up about 3K of memory.

If you leave COMMAND.COM with EXIT, the original environment is restored.

The EXIT command has no effect on the last command processor, which you cannot leave any more.

If you have left the MS-DOS shell with Shift + F9 or with a corresponding menu command, use the EXIT command to return to the Shell.

In many applications you have the possibility of jumping to the operating system. A second command processor is loaded and started. All commands that are entered have no influence on the original environment of the application program. You leave this copy with the EXIT command:

```
EXIT
```

See also: COMMAND

EXPAND

(external)

Expands one or more compressed MS-DOS 5.0 files.

Syntax:	EXPAND [*drive:*][*path*]*filename*[....]]*destination*
drive:	Source drive
path	Source path
filename	Filename of file to be expanded. Wildcards are not permitted. Multiple drives, paths and filenames may be specified.
destination	Specifies the location and/or name of the expanded file or files.

Most of the files supplied on the distribution diskettes are compressed and are automatically expanded when you install MS-DOS 5.0. These compressed files have a filename extension which ends with the underscore character (_). Occasionally it may want to copy a compressed file and manually expand it. For example, if you accidentally delete a DOS command file.

Generally, you will find little use for this command.

FASTOPEN

(external)

Decreases the amount of time required to open frequently used files and directories.

Syntax: FASTOPEN *drive:* [[=]*number*][*drive:* [[=]*number*][....]][/x]

drive:	The hard drive whose filenames are to be cached.
number	The maximum number of filenames to be cached. Values may range from 10 to 999. The default value is 48.

Switches: /x Creates the filename cache in expanded memory.

Description:

When opening a file, DOS searches multiple directories to locate the file. This can be time consuming if you are using a hard drive with many subdirectories. FASTOPEN caches the filenames and their locations. If one of these files is accessed again, it can be located very quickly from the cache.

FASTOPEN works only with hard drives and does not work on networks.

A FASTOPEN command can be defined for any disk drive with up to 24 partitions. FASTOPEN requires 48 bytes for each entry. If the number of files opened exceeds the amount specified by `number`, un-referenced filenames are deleted from the list and the new filenames are inserted in the list accordingly.

Use the INSTALL command with your CONFIG.SYS file to automatically run FASTOPEN when booting your computer. You can also run FASTOPEN from your AUTOEXEC.BAT file.

The following starts FASTOPEN from the CONFIG.SYS file. It caches a maximum of 100 filenames on the C: drive:

```
INSTALL=C:\DOS\FASTOPEN.EXE c:=100
```

If FASTOPEN is to work with multiple hard drives, a corresponding assignment must be made for each drive. Only one FASTOPEN command is allowed. You cannot cache two drives by using two FASTOPEN commands. A single command is necessary. In the following example the access to filenames on drives C: and D: are cached:

```
FASTOPEN c:=100 d:100
```

☞ Do not use disk optimization programs such as Norton's Speed Disk or any similar programs when FASTOPEN is installed. Running a disk optimization program with FASTOPEN installed may cause loss of data.

FC

(external)

Compares two files or sets of files and displays the differences between them.

Syntax: FC [/a][/c][/l][/LB*n*][/n][/t][/w][/*nnnn*][*drive1*:][*path1*]*filename1* [*drive2*:][*path2*]*filename2*
FC /b [*drive1*:][*path1*]*filename1* [*drive2*:][*path2*] *filename2*

Switches:

/a	Displays only the first and last lines for each set of differences.
/b	Performs a binary comparison.
/c	Ignore upper and lowercase letter differences.
/l	Compares files as ASCII text.
/LB*n*	Sets the number of lines for the internal line buffer. The default length of the line buffer is 100 lines. If the files being compared have more than this number of consecutive differing lines, only the first 100 lines are compared.
/n	Displays line numbers during ASCII comparison.
/t	Do not expand tabs to spaces.
/w	Compress white spaces (tabs and spaces) during comparison.
/*nnnn*	Specifies the number of consecutive lines that must match before FC considers the files synchronized again after a mismatch.

Description:

FC compares two files and reports differences between the files.

When used for an ASCII comparison, the differences between the two files are reported in the following order: the name of the first file, the lines that differ between the files and the first line to match in both files. Then the name of the second file is displayed followed by the lines from that file that differ and the first line to match.

The following format is used to report mismatches during a binary comparison:

```
xxxxx: yy zz
```

The value represented by xxxxx specifies the hexadecimal address for the pair of bytes, offset from the start of the file. yy and zz represent the mismatched bytes from the two files being compared.

Wildcards are permitted in filenames.

FDISK

(external)

Configures a hard drive for use with MS-DOS.

Syntax: FDISK

Description:

Hard drives can be divided into several different areas called *partitions*. A hard drive is often pre-partitioned and pre-formatted when you buy your computer. If not, you must prepare the hard drive using the FDISK command.

FDISK may be used to:

- Create a primary MS-DOS partition.
- Create an extended MS-DOS partition.
- Create logical MS-DOS drives.
- Change the active partition.
- Delete a partition.
- Display partition information.
- Select the next hard drive for partitioning on a system with multiple hard drives.

The hard drive can be divided into different partitions of varying sizes. A different operating system can then be assigned to each of these partitions.

Two different types of partitions may be placed on a hard drive: The primary DOS partition and the extended DOS partition. The primary partition contains the information necessary for booting from DOS.

The maximum partition size with MS-DOS 5.0 is 2 gigabytes, which is 2 x 10^9 or 2 billion bytes of storage area.

FDISK is menu-oriented. You select the options through menus instead of by parameters.

FDISK menus allow you to create a partition, change the size of a partition, delete a partition, display all of the information about a partition or select a different hard drive. This option is available only if your PC has access to several hard drives.

☞ It is not possible to create or delete a partition for another operating system. To do this, you must consult the documentation for each different operating system. Also, you cannot change or delete an active partition.

The information about a DOS partition looks like this:

```
                       Display Partition Information
Partition Status  Type   Volume Label  Mbytes  System  Usage
   C: 1      A   PRI DOS  DOS500          76   FAT16    100%



Total disk space is   76 Mbytes (1 Mbyte = 1048576)


Press Esc to continue
```

In this example, the entire drive is available as a DOS partition. If bad sectors (or tracks) are found on the hard drive, they are marked as unusable when the partition is installed. If you define the entire drive as a DOS partition, then you do not have to switch the DOS partition to active. FDISK asks you to insert the system diskette into drive A: to format the hard drive.

It's important that you format the hard drive with the /s (System transfer) switch of the FORMAT command. This copies the operating system files IBMBIO.COM, IBMDOS.COM and COMMAND.COM to the DOS partition after formatting.

The active partition is always the partition that is booted. If the entire hard drive is a DOS partition, it is always active. Otherwise FDISK displays the status of the defined partitions from which you can choose the partition you want to activate.

Setting up an unformatted hard drive

Insert the system diskette in drive A: and boot it. (For an IBM AT, you must use the diagnostics program to indicate that the drive is attached so that you can access it.)

Next use FDISK. To define the entire hard drive as a DOS partition, select menu option 1. FDISK asks if you want to use the entire hard drive as a DOS partition. Answer yes. Next you are asked to insert the system diskette in drive A:. Type the following command to format the hard drive as a system diskette:

```
FORMAT C: /s/v
```

FIND

(external)

Syntax: FIND [/v][/c][/n][/i]*string*[[*drive:*][*path*]*filename*[...]]

string	The character string to be searched for in a file. *string* must be enclosed in quotation marks. Upper and lowercase letters are searched for exactly as entered. (See switch /i below.)
drive: path	The path to the file to be searched. Wildcards are allowed. Multiple filenames may be specified.
filename	The name of the file to be searched.

Switches:

/v	Inverts the search process — the lines which do not contain the specified string are displayed or counted.
/c	The number of lines which contain the search string are simply counted and not displayed. The /c switch disables the /n switch and has higher priority. The number of lines which contain the string are displayed adjacent to the filename.
/n	Numbers the displayed lines. The line number is displayed in square brackets without leading zeros.
/i	Specifies that the search is not case sensitive.

Description:

FIND searches a specified file or group of files for a character string and displays the line containing the string.

To search for a string containing quotation marks, enter each quotation mark as a double quote (""). For example:

```
FIND /c ".said ""You know how to format diskettes."" demo.txt
```

To search an entire file for two specific strings, chain together two FIND commands. For example, to find all of the lines of a Pascal program that contain a BEGIN statement and a comment, type the following:

```
FIND "BEGIN" DEMO.PAS | FIND "(*"
```

See also: MORE, SORT

FOR

(resident)

Performs a specified command for each file in a file list.

Syntax:

To use FOR from the command line, use the following:

```
FOR %variable IN list DO command [parameters]
```

To use FOR in a batch file, use the following:

```
FOR %%variable IN (list) DO command [parameters]
```

%%variable or *%variable*	Represents a replaceable variable.
list	Specifies a list of one or more files. Wildcards are allowed.
command	Specifies the command to be performed for each file.
parameters	The parameters or switches for the specified command.

Description:

FOR performs a specified command for each file in a list of files. IN and DO are not parameters, but are required keywords for the FOR command.

To avoid confusion with batch file parameters, use any character other than the numerals 0 through 9 for *variable*.

list may be a single group of files or several groups of files. Wildcards may be used to specify a file list. Some valid file lists are:

```
(*.DOC)
(*.DOC *.TXT)
CHAPT??.DOC)
```

Example

```
FOR %a IN (*.TXT) DO TYPE %a
```

This command line displays all files, in the current directory, on the screen using the TYPE command.

This line could be used in a batch file by replacing %a with %%a.

FORMAT

(external)

Prepares a diskette or hard drive for use with MS-DOS.

Syntax:

FORMAT [*drive:*][/v[:*label*]][/q][/u][/f:*size*][/b | /s]
FORMAT [*drive:*][/v[:*label*]][/q][/u][/t:*tracks* /n:*sectors*][/b | /s]
FORMAT [*drive:*][/v[:*label*]][/q][/u][/1][/4][/8][/b | /s]
FORMAT [*drive:*][/q][/u][/1][/4][/8][/b | /s]

drive: Specifies the drive containing the diskette or hard drive to be formatted. The drive must be specified.

Switches:

/b	Reserves space on a diskette for system files. (This switch is no longer necessary, but is maintained for compatibility with early versions of DOS.)
/f:*size*	Specifies the size of the floppy diskette. Valid sizes are 160, 180, 320, 360, 720, 1.2, 1.44 and 2.88.
/n:*xx*	Format a disk with *xx* sectors per track. When possible, use the /f switch instead of this switch.
/q	Perform a "safe" (quick) format.
/s	Format the diskette as a system diskette. The two hidden files IO.SYS and MSDOS.SYS are copied to the formatted diskette.
/t:*yy*	Format a disk with *yy* tracks. When possible, use the /f switch instead of this switch.
/u	Perform a complete (unconditional) format.
/v[:*label*]	Assigns a volume label to a diskette or hard drive.
/1	Format the diskette as single-sided. This switch is necessary if you are using single-sided diskettes.
/4	Format a 5.25" diskette with a capacity of 360K in a high density drive. This switch is only relevant when formatting with 1.2 Meg drives.
/8	Format the diskette with eight sectors per track.

Description:

A diskette or hard drive must be formatted before data can be stored on it. Formatting divides the disk into sides, tracks (cylinders) and sectors. In addition, formatting creates a *File Allocation Table* on the new disk. A FAT contains information about a disk's used sectors.

Since formatting erases essentially all data on a diskette or hard drive, a confirmation is required after the program is started. (Actually, MS-DOS 5.0 saves information for recovering an accidentally formatted diskette. See UNFORMAT.) FORMAT automatically assigns a volume serial number to the formatted diskette. This serial number is displayed after the format operation.

The FORMAT command defaults to formatting at the highest density possible. This density depends on the drive's capabilities. For example, a 5 1/4 inch drive capable of writing to 1.2 Meg diskettes will format diskettes at 1.2 Meg, unless you instruct the drive to do otherwise. You can control the format density by using the /4 or /f:*size* switches.

To format your hard drive, make sure that the hard drive is correctly installed and partitioned. (See the FDISK command.) Format the hard drive with:

```
FORMAT c: /s /v
```

When formatting a hard drive, you'll be asked the following as a precaution:

```
WARNING! ALL DATA ON  NON-REMOVABLE DISK
DRIVE C: WILL BE LOST!
Proceed with Format (Y/N)?
```

To continue to format the hard drive, press Y. Pressing N cancels the command.

If the /u switch is not specified, or the disk is not formatted to a different size, MS-DOS performs a "safe" format. This means that the FAT (File Allocation Table) and the root directory are cleared but data is not erased. The UNFORMAT command can be used to recover this information.

Formatting checks to ensure that each sector can store data properly. If a sector is found that cannot store data, it is marked to prevent MS-DOS from using it.

If the /u switch is specified or the size of the diskette is changed, an unconditional format is performed and all data on the disk is erased.

The /n switch specifies the number of sectors per track. The /t switch specifies the number of tracks per side. These switches are alternatives to using other switches - /1, /8, etc.

The /f switch specifies the capacity of the formatted diskette in bytes, K bytes or megabytes. This simplifies using the FORMAT command. All of the allowable formats and the corresponding switches are shown in the following table:

Format	Possible Switches
160K	/f:160 /f:160K
180K	/f:180 /f:180K
320K	/f:320 /f:320K
360K	/f:360 /f:360K
720K	/f:720 /f:720K
1.2 Meg	/f:1200 /f:1200K /f:1.2 /f:1.2 M
1.44 Meg	/f:1440 /f:1440K /f:1.44 /f:1.44M
2.88 Meg	/f:2880 /f:2880 /f:2.88 /f:2.88M

Not all switches are allowed with each drive. The following table provides an overview of the switches.

Drive	Allowable Switches
160/180K	/1 /4 /8 /b /n /t /v /s /f
320/360K	/1 /4 /8 /b /n /t /v /s /f
720K	/b /n /t /v /s /f
1.2 Meg	/b /n /t /v /s /f
1.44 Meg	/b /n /t /v /s /f
2.88 Meg	/b /n /t /v /s /f
hard drive	/b /v /s

Diskette Type	Drive	Tracks	Sectors	tpi	in DOS
5.25" 1s/dd	60K	40	8	48	1.0
5.25" 1s/dd	180K	40	9	48	2.0
5.25" 2s/dd	320K	40	8	48	1.1
5.25" 2s/dd	360K	40	9	48	2.0
5.25" 2s/dd	1.2 Meg	80	15	96	3.0
3.5" 2s/dd	720K	80	9	135	3.3
3.5" 2s/hd	1.44 Meg	80	18	270	3.3
3.5" 2s/hd	2.88 Meg	80	36	540	5.0

Return Values for Batch File Processing (error level):

Code	Meaning
0	FORMAT was successful, no errors
3	stopped with Ctrl+C
4	error during formatting
5	negative response to message: "Proceed with Format (Y/N?)"

GOTO

(resident)

Jump to a labeled line during execution of a batch file.

Syntax:	GOTO *label*
label	Specifies the line in a batch program to which MS-DOS should jump.

GOTO jumps to a line, identified with a label, during a batch program. The batch file continues executing commands at the line following the label.

Labels in batch files begin with a colon and can contain spaces, but cannot include other separators, such as colons or equal signs. Only the first eight characters of a label are used.

GOTO is often used to perform conditional operations in conjunction with IF.

For more detailed information on the use of GOTO in batch files, refer to Chapter 14.

GRAFTABL

(external)

Sets up MS-DOS to display an extended character set in graphics mode.

Syntax: GRAFTABL [*nnn*]
GRAFTABL [/status]

Switches:		
	nnn	Specifies the code page number which defines the appearance of the characters.
	/status	Identifies the active code page.

Description:

GRAFTABL affects only the appearance of external characters (ASCII characters 128-255) in the specified code page, but does not change the code page. To modify the code page, use MODE or CHCP.

The following shows each valid code page ID number and its country or language.

Code	Meaning
437	USA (standard)
850	Multilingual (Latin I)
852	Slavic (Latin II)
860	Portuguese
863	Canadian-French
865	Nordic

The /status switch displays the active code page.

If GRAFTABL is entered without switches the following information is displayed:

```
C:>GRAFTABL
Previous Code Page: None
Active Code Page: 437
```

GRAFTABL resides in memory after being run and requires about 1K of conventional memory.

Returns Values for Batch File Processing (error level):

Code	Meaning
0	GRAFTABL was installed as resident
1	An existing table was overwritten
2	A file error occurred
3	An incorrect switch was specified, GRAFTABL terminated
4	Incorrect DOS version, DOS 5.0 is required

See also: CHCP, GRAPHICS, MODE, NLSFUNC

GRAPHICS

(external)

Loads a program to print graphics on a printer when using a color or graphics adapter.

Syntax: GRAPHICS [*type*] [*drive:*][*path*][*filename*][/r][/b][/LCD]
[/PRINTBOX:STD | PRINTBOX:LCD]

drive:path filename
Specifies the path and filename containing printer information.

Switches: *type*
Specifies the type of printer. See the table below for a list of valid printer types.

[/r] Prints the image with white characters on a black background, as it appears on your screen.

[/b] This switch is valid for COLOR4 and COLOR8 printers. The background is printed in color.

[/LCD]
Uses the liquid crystal (LCD) aspect ratio instead of CGA aspect ratio.

/PRINTBOX:STD | PRINTBOX:LCD
Selects the size of the print-box, either std or lcd.

Description:

GRAPHICS loads a program to print graphics. The GRAPHICS command supports CGA, EGA and VGA display modes.

type is one of the following printers:

Mode	Printer
color1	IBM personal computer color printer with black ribbon
color4	IBM personal computer color printer with red, green, blue and black color ribbon
color8	IBM personal computer color printer with cyan, magenta, yellow, black ribbon
hpdefault	Any Hewlett-Packard PCL printer
deskjet	Hewlett-Packard DeskJet printer
graphics	IBM personal computer compact printer or a compatible
graphicswide	Graphic printer with wide paper (11 inches)
laserjet	Hewlett-Packard LaserJet printer
laserjetii	Hewlett-Packard LaserJet II printer
paintjet	Hewlett-Packard PaintJet printer
quietjet	Hewlett-Packard QuietJet printer
ruggedwriter	Hewlett-Packard RuggedWriter printer
ruggedwriterwide	Hewlett-Packard RuggedWriter wide printer
thermal	IBM PC convertible Thermal Printer
thinkjet	Hewlett-Packard ThinkJet printer

There is extensive information about the supported graphic printer in a file named GRAPHICS.PRO.

See also: CHCP, GRAFTABL, MODE, NLSFUNC

HELP

(external)

Displays on screen help for MS-DOS commands.

Syntax: HELP [*command*]
or
[*command*] /?

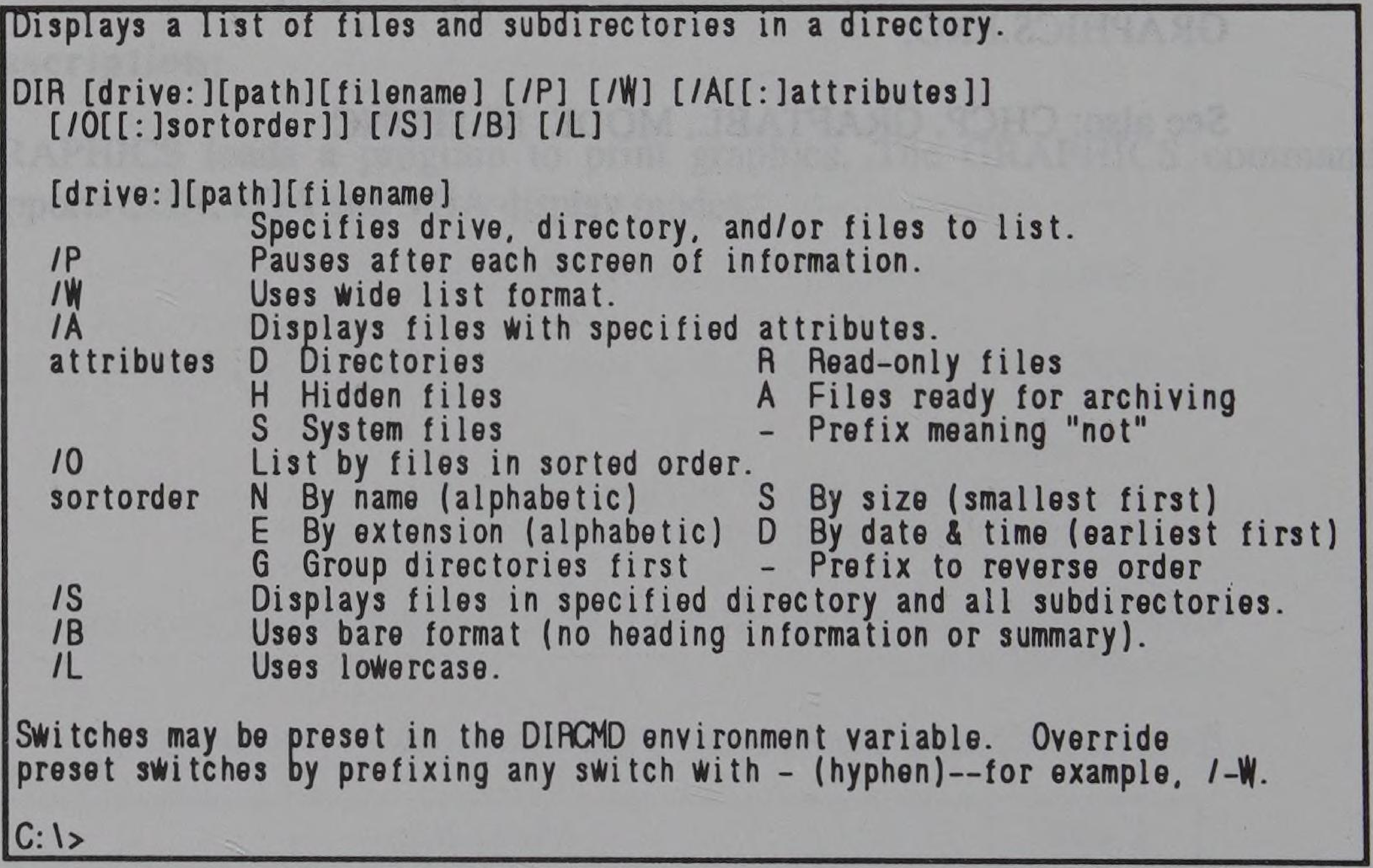
command	Specifies the name of the command about which information is required. If no command is specified, HELP displays every command and a brief summary of its function.

The HELP command displays on screen help for every MS-DOS command. All parameters and switches are listed. A brief description of the command's function and a description of the effect of the various switches are displayed.

Example

The following is displayed by typing:

```
HELP DIR
or
DIR /?
```



```
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/P] [/W] [/A[[:]attributes]]
  [/O[[:]sortorder]] [/S] [/B] [/L]

  [drive:][path][filename]
              Specifies drive, directory, and/or files to list.
  /P          Pauses after each screen of information.
  /W          Uses wide list format.
  /A          Displays files with specified attributes.
  attributes  D  Directories                R  Read-only files
              H  Hidden files               A  Files ready for archiving
              S  System files               -  Prefix meaning "not"
  /O          List by files in sorted order.
  sortorder   N  By name (alphabetic)       S  By size (smallest first)
              E  By extension (alphabetic)  D  By date & time (earliest first)
              G  Group directories first    -  Prefix to reverse order
  /S          Displays files in specified directory and all subdirectories.
  /B          Uses bare format (no heading information or summary).
  /L          Uses lowercase.

Switches may be preset in the DIRCMD environment variable.  Override
preset switches by prefixing any switch with - (hyphen)--for example, /-W.

C:\>
```

IF

(external)

Performs conditional processing in batch files.

Syntax:
IF [NOT] ERRORLEVEL *number command*
IF [NOT] *string1==string2 command*
IF [NOT] EXIST *filename command*

[NOT]	Specifies that the command should be executed only if the condition is false.
ERRORLEVEL *number*	Specifies a true condition only if the previous program run by COMMAND.COM returned an exit code equal to or greater than the value specified by *number*.
command	The command to execute if the preceding condition is met.
string1==string2	The condition is true if *string1* and *string2* are the same. These values can be literal strings or batch variables.
EXIST *filename*	If *filename* exists, the condition is true.

In a batch file, IF tests for specified conditions and performs or skips commands based on the outcome of the test. If the text condition is met, the command is performed; otherwise it skips to the next line of the batch file.

The double equal sign (==) tests for equality.

IF can also be combined with other conditions:

IF NOT Condition

Here, if the condition is met, then the batch file execution continues with the next line. If it's not met, then the rest of the line is executed.

IF EXIST filename

Checks to see if the specified file exists. If it does, then the rest of the command line is executed. Otherwise, processing continues with the next line. A variation of this command is IF NOT EXIST. In this case, the rest of the line is processed only if the specified file does not exist.

IF ERRORLEVEL number

Checks for an error number. When certain MS-DOS commands encounter problems or errors, they return an error number to a batch file. The batch file can determine the error number by using the ERRORLEVEL parameter. If there is no error, ERRORLEVEL is 0. Otherwise the various MS-DOS commands return different values of ERRORLEVEL depending on the specific problem or error.

The commands in the IF ERRORLEVEL line are executed if the ERRORLEVEL returned is greater than or equal to the specified number. Otherwise, it skips to the next line.

JOIN

(external)

Redirects a disk drive to a directory on another disk drive.

Syntax: JOIN [*drive1*:][*drive2*:][*path*][/d]
JOIN *drive1*: /d

drive1:	Specifies the disk drive or hard drive to be joined to a different drive and directory.
drive2:	Specifies the drive to receive the combined capacity.
path	A subdirectory of *drive2*; it must be empty. It must also be a directory other than the root directory. If the directory does not exist, MS-DOS tries to create it.

Switches: /d Cancels any previous JOIN commands for the specified drive.

JOIN lets you access *drive1:* as though it were the subdirectory *drive2:path*. When you join *drive1:*, it becomes accessible only through *drive2:path*. Trying to access *drive1:* produces the error message: "Invalid drive specification".

JOIN lets you run certain software applications that don't support a hard drive. You can set up a directory (called \HD in this example) that points to the hard drive C:

```
JOIN A: C:\HD
```

Directory \HD is created on drive C: if it does not already exist. Type the following line to display this directory of drive A:, which can only be accessed as C:\HD:

```
DIR \HD
```

See also: ASSIGN, SUBST

KEYB

(external)

Sets up the keyboard for a specific language.

Syntax: KEYB [*keyboard_code*[,*code_page*[,[*drive:*][*path*]*filename*]]][/e][/ID:id]

keyboard_code
The two-character codes used to define the country-specific keyboard.

code_page
The code page. The default is the standard character table.

filename
The keyboard definition file, usually the file KEYBOARD.SYS. This parameter is required if it is not in the root directory.

Switches: /e Specifies that an enhanced keyboard is installed.

/ID:*id*
Defines the keyboard used in countries that have different models. This switch is only necessary in countries that have more than one keyboard layout for the same language (France, Italy and the United Kingdom).

Description:

The following table shows valid values for *keyboard_code*, the *code_page* and *id* (when necessary):

Country	Keyboard code	Code Page	ID
Australia	US	850,437	
Belgium	BE	850,437	
Canada (Fr.)	CF	850,863	
Canada, Engl.	US	850,437	
Denmark	DK	850,865	
Finland	SU	850,437	
France	FR	850,437	120,189
Germany	GR	850,437	
Hungary	HU	852,850	
Italy	IT	850,437	141,142
Latin America	LA	850,437	
Netherlands	NL	850,437	
Norway	NO	850,865	
Poland	PL	852,850	
Portugal	PO	850,860	
Spain	SP	850,437	
Sweden	SV	850,437	
Switzerland (French)	SF	850,437	
Switzerland (German)	SG	850,437	
United Kingdom	UK	850,437	166,168
United States	US	850,437	
Yugoslavia	YU	852,850	

There are different keyboard arrangements available for Great Britain, France and Italy. The value of ID specifies the keyboard model required.

The code page specifies the character set table that should be used to display the character on the screen when a given key is pressed. The supported character set tables are:

Character table	
437	USA (standard IBM character set)
850	multi-language
860	Portugal
863	French Canadian and French
865	Norway

Not every character set table can be used with each keyboard arrangement. The following table lists which keyboard driver may be combined with the respective character set tables:

Set	Country
437	BE, BR, SU, FR, GR, IT, LA, NL, SP, SV, SF, SG, UK, US
850	BE, BR, CF, CZ, SL, DK, SU, FR, GR, HU, IT, LA, NL, NO, PL, PO, SP, SV, SF, SG, UK, US, YU
852	CZ, SL, HU, PL, YU
860	PO
863	CF
865	DK, NO

To use a Portuguese keyboard arrangement, type the following command:

```
KEYB po, 860
```

If the file KEYBOARD.SYS is not in the main directory, specify the path:

```
KEYB po, 860, c:\dos
```

If you don't want to define a character set table, you can omit the second parameter:

```
KEYB po, ,c:\dos\keyboard.sys
```

If KEYB is entered without parameters, the current settings are displayed:

```
KEYB
Actual keyboard code: UK Character set table:   437
Active character set table not usable from      unit CON
```

See also: CHCP, MODE, NLSFUNC, SELECT

LABEL

(external)

Creates, changes or deletes the volume label of a disk.

Syntax: LABEL [*drive:*][*name*]

drive:	The drive that contains the disk to be named. If omitted, the current drive is assumed.
name	The volume label.

Description:

LABEL assigns a volume label to a disk. The volume label may contain up to 11 characters.

If *name* is omitted, the following DOS message prompts you for a new volume label:

```
Volume label (11 characters, ENTER for none)?
```

To change the volume label, type the new volume label and press Enter.

To delete the previous volume label, press Enter:

```
LABEL C:harddisk
LABEL A:
Volume label (11 characters, ENTER for none)?
```

If a volume label exists, pressing Enter without entering text at the "Volume label" prompt displays:

```
Delete current volume label (Y/N)?
```

LH

(resident)

Loads a program into the upper memory area of the computer.

Syntax: LH [*drive:*][*path*]*filename* [*parameters*]

parameters	Any command line information required by the program to be loaded.

Description:

LH loads a program into the upper memory area leaving more room in conventional memory for other programs.

Before you can use LH, you must install the HIMEM.SYS driver and include the DOS=UMB command in your CONFIG.SYS file.

If there is insufficient space to load a program into the upper memory area, MS-DOS loads the program into conventional memory. There is no indication from MS-DOS as to which memory area was used.

For convenience, you can use the LH command from your AUTOEXEC.BAT file.

See also: LOADHIGH

LOADFIX

(external)

Loads a program into the memory area above 64K and runs the program.

Syntax: LOADFIX [*drive:*][*path*]*filename*

[*drive:*][*path*]*filename*
Complete path and name of file to load and execute.

MS-DOS 5.0 is able to "free" certain areas of conventional memory. Some programs won't run properly, especially ones which expect to be loaded at a specific memory location. LOADFIX loads and runs programs that produce the "Packed file corrupt" error message.

LOADFIX can also load some programs which fail to load in a network environment.

LOADHIGH

(resident)

Loads a program into the upper memory area of the computer.

Syntax:	LOADHIGH [*drive:*][*path*]*filename* [*parameters*]
parameters	Any command line information required by the program to be loaded.

Description:

LOADHIGH loads a program into the upper memory area leaving more room in conventional memory for other programs.

Before you can use LOADHIGH, you must install the HIMEM.SYS driver and include the DOS=UMB command in your CONFIG.SYS file.

If there is insufficient space to load a program into the upper memory area, MS-DOS loads the program into conventional memory. There is no indication from MS-DOS as to which memory area was used.

For convenience, you can use the LOADHIGH command from your AUTOEXEC.BAT file.

MD *(resident)*

Creates a new directory.

Syntax: MD [*drive:*][*path*]

drive:	The drive on which the new directory is to be created.
path	Name and location of the new directory.

Description:

MD creates a new subdirectory on *drive:*.

If the new subdirectory is to appear in the root directory, *path* must be preceded by a backslash symbol (\). If this symbol is omitted, the subdirectory is created as a subdirectory of the current directory.

☞ Don't use MD when any of the ASSIGN, JOIN or SUBST commands are active.

See also: MKDIR

MEM

(external)

Displays information about memory usage in your system.

Syntax: MEM [/debug | /program | /classify]

Switches: /program or /p

Displays information about the memory usage of currently loaded programs.

/debug or /d

Displays information about the memory usage of programs, drivers, internal buffers, etc.

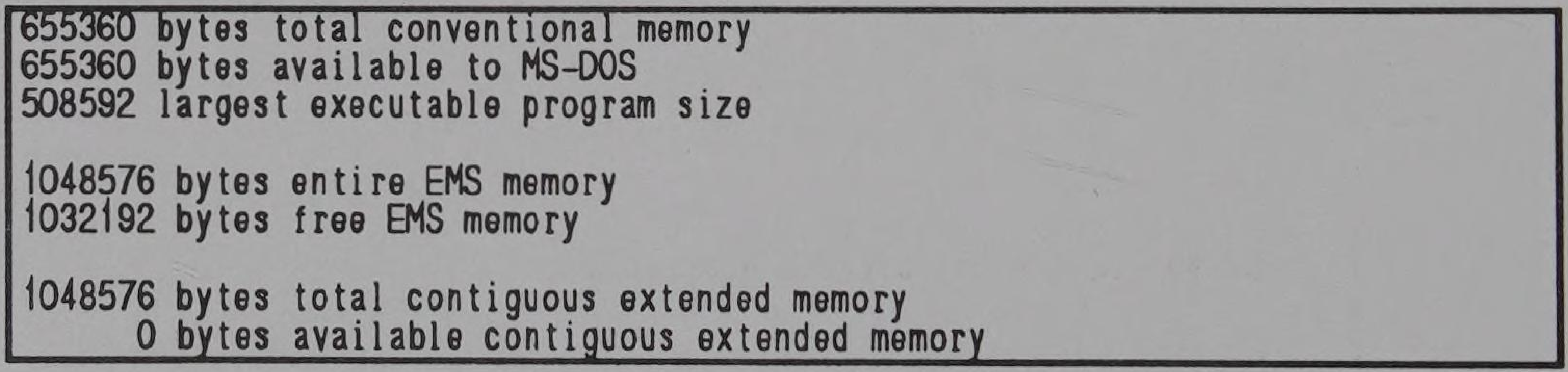
/classify or /c

Displays summarized information about the memory usage of programs and unused space.

Description:

The MEM command displays information about the memory of your computer. Information about the extended and expanded memory as well as the main memory is given.

Typing MEM with no switches displays a screen similar to the following:



```
655360 bytes total conventional memory
655360 bytes available to MS-DOS
508592 largest executable program size

1048576 bytes entire EMS memory
1032192 bytes free EMS memory

1048576 bytes total contiguous extended memory
      0 bytes available contiguous extended memory
```

Entering MEM /p will display information similar to the following:

```
Address     Name        Size        Type
-------     --------    ------      ------
000000                  000400      Interrupt Vector
000400                  000100      ROM Communication Area
000500                  000200      DOS Communication Area

000700      IO          000A60      System Data

001160      MSDOS       0013D0      System Data

002530      IO          0079F0      System Data
              HIMEM     000B50        DEVICE=
              SETVER    000190        DEVICE=
              EMM386    0020D0        DEVICE=
              MOUSE     0039E0        DEVICE=
                        0005D0        FILES=
                        000100        FCBS=
                        000200        BUFFERS=
                        0001C0        LASTDRIVE=
                        000740        STACKS=
009F30      MSDOS       000040      System Program

009F80      MSDOS       000060      -- Free --
009FF0      FASTOPEN    001A90      Program
00BA90      COMMAND     000940      Program
00C3E0      MSDOS       000040      -- Free --
00C430      COMMAND     000100      Environment
00C540      MSDOS       0000D0      -- Free --
00C620      DOSKEY      001020      Program
00D650      CAPTURE     0000E0      Environment
00D740      GRAFTABL    0004A0      Program
00DBF0      CAPTURE     006130      Program
013D30      MEM         0000E0      Environment
013E20      MEM         0176F0      Program
02B520      MSDOS       074AD0      -- Free --

    655360 bytes total conventional memory
    655360 bytes available to MS-DOS
    573904 largest executable program size

    655360 bytes total EMS memory
    262144 bytes free EMS memory

   1441792 bytes total contiguous external memory
         0 bytes available contiguous external memory
   1015808 bytes available XMS memory
           MS-DOS resident in High Memory Area
```

Total Conventional Memory is the amount of memory on your computer up to the first 640K.

Available to MS-DOS is the amount of conventional memory MS-DOS has for operating the computer.

Largest Executable Program Size is the largest contiguous block of conventional memory available for a program.

Total EMS memory is the amount of expanded memory installed.

Total contiguous external memory is the amount of memory beyond the 1 Meg limit.

Available contiguous external memory is the external memory available for the Interrupt 15h interface.

Available XMS memory is the memory that is being managed by an external memory manager such as HIMEM.SYS.

Entering MEM /d will display information similar to the following:

```
Address     Name       Size      Type
-------     --------   ------    ------
000000                 000400    Interrupt Vector
000400                 000100    ROM Communication Area
000500                 000200    DOS Communication Area

000700      IO         000A60    System Data
              CON                  System Device Driver
              AUX                  System Device Driver
              PRN                  System Device Driver
              CLOCK$               System Device Driver
              A: - C:              System Device Driver
              COM1                 System Device Driver
              LPT1                 System Device Driver
              LPT2                 System Device Driver
              LPT3                 System Device Driver
              COM2                 System Device Driver
              COM3                 System Device Driver
              COM4                 System Device Driver

001160      MSDOS      0013D0    System Data

002530      IO         0079F0    System Data
             HIMEM     000B50     DEVICE=
              XMSXXXX0              Installed Device Driver
             SETVER    000190     DEVICE=
              SETVERXX              Installed Device Driver
             EMM386    0020D0     DEVICE=
              EMMXXXX0              Installed Device Driver
             MOUSE     0039E0     DEVICE=
              MS$MOUSE              Installed Device Driver
                       0005D0     FILES=
                       000100     FCBS=
                       000200     BUFFERS=
                       0001C0     LASTDRIVE=
                       000740     STACKS=
009F30      MSDOS      000040    System Program

009F80      MSDOS      000060    -- Free --
009FF0      FASTOPEN   001A90    Program
00BA90      COMMAND    000940    Program
00C3E0      MSDOS      000040    -- Free --
00C430      COMMAND    000100    Environment
00C540      MSDOS      0000D0    -- Free --
00C620      DOSKEY     001020    Program
00D650      CAPTURE    0000E0    Environment
00D740      GRAFTABL   0004A0    Program
00DBF0      CAPTURE    006130    Program
013D30      MEM        0000E0    Environment
013E20      MEM        0176F0    Program
02B520      MSDOS      074AD0    -- Free --

  655360 bytes total conventional memory
  655360 bytes available to MS-DOS
  573904 largest executable program size

Handle      EMS Name   Size
-------     --------   ------
     0                 060000

  655360 bytes total EMS memory
  262144 bytes free EMS memory

 1441792 bytes total contiguous external memory
       0 bytes available contiguous external memory
 1015808 bytes available XMS memory
         MS-DOS resident in High Memory Area
```

Entering MEM /c will display information similar to the following:

```
Conventional Memory :

  Name              Size in Decimal       Size in Hex
-------------     ---------------------   -------------
  MSDOS              14144    ( 13.8K)        3740
  HIMEM               2896    (  2.8K)         B50
  SETVER               400    (  0.4K)         190
  EMM386              8400    (  8.2K)        20D0
  MOUSE              14816    ( 14.5K)        39E0
  FASTOPEN            6800    (  6.6K)        1A90
  COMMAND             2624    (  2.6K)         A40
  DOSKEY              4128    (  4.0K)        1020
  CAPTURE            25104    ( 24.5K)        6210
  GRAFTABL            1184    (  1.2K)         4A0
  FREE                  96    (  0.1K)          60
  FREE                  64    (  0.1K)          40
  FREE                 208    (  0.2K)          D0
  FREE              574144    (560.7K)        8C2C0

Total  FREE :       574512    (561.0K)

Total bytes available to programs :                         574512   (561.0K)
Largest executable program size :                           573904   (560.5K)

    655360 bytes total EMS memory
    262144 bytes free EMS memory

   1441792 bytes total contiguous external memory
         0 bytes available contiguous external memory
   1015808 bytes available XMS memory
           MS-DOS resident in High Memory Area
```

See also: CHKDSK

MIRROR *(external)*

Saves information about disks for use by the UNFORMAT and UNDELETE commands.

Syntax: MIRROR [*drive:* [...]] [/1][/*tdrive:*[*-entries*][...]]
MIRROR [/u]
MIRROR [/partn]

drive: Specifies the drive about which information will be saved.

Switches:

/1 Save only the most recent information about the disk. If this switch is omitted, MIRROR makes a backup copy of the existing disk information file before saving the current information.

/t*drive*[- *entries*]
Loads a program for recording information (deletion tracking file) to be used by the UNDELETE command. The required parameter, *drive*, specifies the disk to be mirrored. The *entries* parameter specifies the maximum number of entries to be contained in the deletion tracking file (PCTRACKR.DEL). The value must be in the range of 1 - 999.

/u Removes the program for recording UNDELETE information (deletion tracking file) from memory.

/partn
Saves hard drive partitioning information.

MIRROR saves a copy of the File Allocation Table (FAT) and the root directory of the disk of the specified drive. This information is used by the UNFORMAT command to rebuild a disk which has accidentally been formatted.

This information should be saved frequently. You can add a MIRROR command to your AUTOEXEC.BAT file.

You can also save information about the partitioning of your hard drive. Every hard drive has at least one partition and the information about the partition is stored in a special disk partition table. If this table becomes corrupted, MS-DOS can't access the hard drive.

The partition information can be saved by using MIRROR with the /PARTN switch. This switch creates a file called PARTNSAV.FIL. This file is used by the UNFORMAT command to reconstruct the partition table. This file (PARTNSAV.FIL) should not be saved on your hard drive.

See also: UNDELETE, UNFORMAT

MKDIR

(resident)

Creates a new directory.

Syntax:	MKDIR [*drive:*]*path* MD [*drive:*]*path*
drive:	The drive on which the new directory will be created. If omitted, the current drive is assumed.
path	The name and location of the new directory.

Description:

MKDIR or MD creates a new subdirectory on *drive:*.

If the new subdirectory is to appear in the root directory, *path* must be preceded by a backslash symbol (\). If this symbol is omitted, the subdirectory is created as a subdirectory of the current directory.

☞ Don't use MKDIR or MD when any of the ASSIGN, JOIN or SUBST commands are active.

See also: CD, RD

MODE (Set Display Mode) *(external)*

This form of MODE selects the display mode or reconfigures the active display adapter.

Syntax: MODE [*display_adapter*][,*shift*[,t]]
MODE [*display_adapter*][,n]
MODE CON[:][cols=*c*][lines=*n*]

display_adapter
This is one of the following:

Code	Meaning
mono	Sets the display for the monochrome adapter (80-character text)
40	Sets the screen width to 40 characters per line for the CGA
80	Sets the screen width to 80 characters per line for the CGA
co40	Sets display for CGA using 40 columns
co80	Sets display for CGA using 80 columns
bw40	Sets the display for the CGA using 40 columns in monochrome
bw80	Sets the display for the CGA using 80 columns in monochrome

shift	Shift the display to the right or left. An 80-column display is shifted by two columns and a 40-column display is shifted by one column. Valid values for *shift* are L (for left) and R (for right).
t	Enables aligning screen by using a test pattern.
CON:	Refers to monitor.
cols=	Specifies the number of characters per column.
lines=	Specifies number of lines that can be displayed on the screen.

Description:

MODE sets the CGA for 40- or 80-column display in either monochrome or color.

The optional parameters allow for the adjustment of the display.

MODE (Configure Printer) *(external)*

This form of MODE configures a printer connected to the parallel printer port (generally, LPT1).

Syntax:

```
MODE parallelport[:][COLS=width][,[LINES=lines]
[RETRY=r]
```

parallelport	The printer port device. It can be PRN, LPT1, LPT2 or LPT3.
width	The number of print characters per line. It can be either 80 or 132. The default value is 80.
lines	The number of lines per inch for vertical spacing. It can be either 6 or 8. The default value is 6.

Switches: R Specifies the retry action to take if a time-out occurs. This parameter causes a part of MODE to remain resident in memory. See below for a list of valid values for *r*.

Description:

MODE sets the printer device for 80 or 132 column line widths and/or 6 or 8 lines per inch of vertical spacing.

Valid values for Retry

e	Return an error from a status check of a busy port
b	Return "busy" from a status check of a busy port
p	Continue retrying until printer accepts output
r	Return "Ready" from a status check of a busy port
n	Take no retry action (default value)

You can interrupt a timeout loop by pressing Ctrl+C.

MODE (Configure Serial Port) *(external)*

This form of MODE sets the operating characteristics of the serial port.

Syntax:

MODE COM*n*[:][*baud*],[*parity*],[*data*],[*stop*],[*retry*]
MODE COM*n*[:][baud=*b*] [parity=*p*]
[data=*d*][stop=*s*][retry=*r*]

COM*n*	Specifies the number of the asynchronous communications port. This can be COM1 through COM4.
baud	The baud rate. *baud* can be 110, 150, 300, 600, 1200, 2400, 4800, 9600 or 19,200. You may abbreviate *baud* to its first two digits.
parity	The type of parity checking. *parity* can be E, O or N (even, odd, no parity). If omitted, E is assumed.
data	The number of data bits. *bits* can be in the range of 5 through 8. If this parameter is omitted, 7 is assumed.
stop	The number of stop bits. *stop* can be 1, 1.5 or 2. If omitted and *baud* is 110, than 2 is assumed. Otherwise 1 is assumed.
retry	Specifies the retry action if a time-out occurs when port is busy.

Description:

MODE sets the asynchronous communications port parameters.

Valid retry actions

e	Return an error from a status check of a busy port
b	Return "busy" from a status check of a busy port
p	Continue retrying until printer accepts output
r	Return "Ready" from a status check of a busy port
n	Take no retry action (default value)

MODE (Redirect Printing) *(external)*

This form of MODE redirects all output from the parallel port to the serial port.

Syntax: MODE *parallel_port*[:]= *commport*[:]

parallel_port
The printer port device. This device can be LPT1, LPT2 or LPT3.

commport
The asynchronous communication port (the Serial Port). Valid ranges for *commport* are 1 through 4. The serial port must be initialized with MODE *commport* before redirecting the output to the *parallel port*.

Description:

MODE redirects the data from the printer port to the asynchronous communications port (the serial port).

MODE (Set Device Code Pages) *(external)*

This form of MODE prepares, selects, refreshes or displays the numbers of the code pages for parallel printers or the console.

Syntax:
MODE *device* CODEPAGE PREPARE= (*yyy* [...])
[*drive:*][*path*][*filename*]
MODE *device* CODEPAGE SELECT=*yyy*
MODE *device* CODEPAGE [/status]
MODE *device* CODEPAGE refresh*

device	One of the devices CON, LPT1, LPT2, LPT3 or PRN.
yyy	Number of the code page to prepare. 437 United States 850 Multilingual (Latin I) 852 Slavic (Latin II) 860 Portuguese 863 Canadian-French 865 Nordic
drive:path filename	The complete path parameters of the file containing the code page information. The system diskettes contain these parameters with the file extension .CPI.
/status	Returns status of the current code pages selected or prepared for the specified device.
refresh	Reinstates the prepared code pages if they are lost as a result of a hardware problem or other error.

Description:

The MODE command handles code page activity for *device*. The keyword CODEPAGE (abbreviated CP) is required.

You prepare a device driver for a character set table with the PREPARE parameter.

```
MODE lpt1 codepage prep=((437,860) 4201.cpi)
MODE lpt1 codepage prep=(437 4201.cpi)
```

Following the PREPARE keyword (abbreviation PREP) you can specify a character set table as a file. The following files are on the system diskette:

```
4201.CPI   IBM PC graphic printer II
4208.CPI   IBM Proprinter, IBM Proprinter XL
5202.CPI   IBM PC Quietwriter III Model 5202
EGA.CPI    Enhanced graphics adapter (EGA) or IBM PS/2
LCD.CPI    IBM PC Liquid Crystal display
```

A character set table can be selected after this definition at any time. Select the next set with the SELECT keyword. The following example selects the character set table 860 for the LPT1 device driver:

```
MODE lpt1 codepage select=860
```

When you want to know the current status of a device driver, use the /status switch to display the current character set table and a driver list:

```
MODE lpt1 codepage /status
```

The REFRESH command reactivates the last defined character set table of a device driver. For example, if the printer has been switched off, reactivate it as follows:

```
MODE lpt1 codepage refresh
```

Programming the keyboard

```
MODE CON[:] RATE=repetition DELAY=delay
```

When you hold down a key on the keyboard of your computer, the corresponding character is repeated after .5 seconds with a repeat frequency of 10 characters per second. This operation is called TYPEMATIC. With old AT keyboards the delay and the repeat frequency can be programmed. The repeat frequency should be set to 32 characters per second and the delay to .25 seconds:

```
MODE con rate=32 delay=1
```

If the keyboard on your computer cannot be programmed, you'll receive the following error message:

```
Function is not supported by this system
```

Output of device status

```
MODE [device] [/STA[TUS]]
```

The MODE command can influence components of the input/output devices. MODE can display a device status list. To do this, give the MODE command without parameters.

See also: CHCP, GRAPHICS, GRAFTABL, KEYB, NLSFUNC

MORE

(external)

Displays one screen of output at a time.

Syntax: MORE < [*drive:*][*path*]*filename*
command | MORE

[*drive:*][*path*]*filename*
Displays the name and path of the file to be used for storing the temporary file used by MORE.

command Any MS-DOS command.

Description:

MORE is used to prevent the screen from scrolling. It temporarily halts screen output of lengthy files before information scrolls off the screen.

MORE is a filter command that is used in conjunction with other DOS commands such as TYPE or DIR.

For example, to display the contents of a large subdirectory, the following command line could be used:

```
DIR | MORE
```

When one full screen is displayed, DOS waits for you to press any key, except Ctrl+Pause (also referred to as the *Break* key), to display the next screen. This key combination terminates the command.

You can redirect input to MORE with the < symbol:

```
MORE < file.txt
```

Note that a space separates MORE from the < symbol. When all but the last line of the screen is filled, the following message appears:

```
--More--
```

Press any key to display the next full screen of data.

See also: DIR, FIND, SORT, TYPE

NLSFUNC

(external)

Loads information to support extended country-specific code-page switching.

Syntax: NLSFUNC [*drive:*][*path*] *filename*

[*drive:*][*path*] *filename*
The name of a file containing country-specific information. By default this is the file COUNTRY.SYS in the root directory.

Description:

NLSFUNC loads country-specific information used by CHCP, MODE CP, KEYB and other device drivers.

If *filename* is omitted, the file specified by the country command in the CONFIG.SYS file is used. If the country command is not in the CONFIG.SYS file, the file COUNTRY.SYS is used.

See also: CHCP, MODE

PATH

(resident)

Sets the search path for executable files.

Syntax:

```
PATH [[drive: ][path][;[drive: ]path]...]
PATH ;
PATH
```

drive: path

The drive and path of the directories to be searched. Multiple drives and directories are searched in the order they appear. If *drive: path* is omitted, the current search path is displayed.

Description:

PATH specifies the directory or directories that are to be searched for commands or batch files not found in the current directory. The search path is defined as the directories specified by the parameters *drive:path.* Multiple parameters are separated by a semicolon (;).

Each use of the PATH command replaces any previous search path.

The PATH command with only a semicolon (;) clears the search path.

The maximum length of the PATH command is 127 characters.

See also: APPEND, SET

PAUSE

(resident)

Suspends processing of a batch program.

Syntax: PAUSE

PAUSE temporarily suspends the processing of a batch file and displays the message: "Press any key to continue...".

It is often used to allow the user to respond to a prompt or condition, such as to insert a diskette into the disk drive. Processing of the batch file continues when a key is pressed.

PRINT

(external)

Prints a text file in background mode while processing of other commands continues.

Syntax: PRINT[/d:*device*][/b:*buffersize*][/u:*ticks1*][/m:*ticks2*][/s:*ticks3*][/q:*qsize*][/t][[*drive*:][*path*] *filename*[...]][/c][/p]

/d:*device*	Specifies the printer device. Valid devices are PRN:, LPT1:, LPT2:, LPT3:, COM1: and COM2:. If *device* is omitted, PRN is assumed.
/b:*buffersize*	Specifies the size of the memory area reserved for storing data before it is sent to the printer. The default buffer size is 512 bytes. The maximum size is 16384 bytes.
/u:*ticks1*	Specifies the maximum number of timer ticks that PRINT will wait while the printer is busy. Ticks occur about 18 times per second. The valid range is 1 - 255. If this parameter is omitted, 1 is assumed.
/m:*tick2*	Specifies the number of timer units that PRINT can take to send a character to the printer. If this parameter is omitted, 2 is assumed.
/s:*tick3*	Specifies the number of clock ticks that the MS-DOS scheduler allocates for background printing. The valid range is 1 to 255. If this parameter is omitted, 8 is assumed.
/q:*qsize*	Specifies the maximum number of files in the print queue (4 to 32). If this parameter is omitted, 10 is assumed.

Switches:

/t	Removes all files from the print queue.
/c	Removes a specified file from the print queue.
drive: pathname	The filename to be added or deleted from the print queue. Wildcards are allowed.
/p	Adds preceding and subsequent filenames to the print queue.

The above parameters can be specified only the first time PRINT is used.

Description:

PRINT prints a series of files to the printer while you are performing other tasks. This is known as *print spooling*.

You cannot send any other output to the printer while PRINT is running.

PRINT automatically sends a form feed after a file has been printed.

The advantage of using the PRINT command is that you can define multiple files to be printed and they are all printed in sequence.

The following defines a print queue for LPT1, a queue size of 32 files and a buffer of 2048 bytes:

```
PRINT /d:LPT1 /q:32 /b:2048 /s:10 /m:4 /u:3
```

The following command prints all of the files with an extension of .TXT, as well as the file READ.ME. The file OLD.TXT is removed from the print queue:

```
PRINT *.TXT /p OLD.TXT /c READ.ME /p
```

PROMPT

(resident)

Defines the appearance of the MS-DOS system prompt.

Syntax: PROMPT [*string*]

string	The new system prompt. If *string* is omitted, the standard system prompt is assumed.

Description:

Normally the system prompt appears as the current drive identifier followed by the greater than symbol (e.g., A:>). The PROMPT command changes the appearance of the system prompt.

To define the appearance of the prompt, you can include the PROMPT command in an AUTOEXEC.BAT file. The *string* parameter is composed of printable characters and/or the following codes. These codes can be used anywhere in the string and as often as necessary:

Code	Meaning
$b	Pipe symbol (|)
$d	System date
$e	Escape character (01bh)
$g	Greater than symbol (>)
$h	Backspace deletes the previous character
$l	Less than symbol (<)
$n	Current drive
$p	Current drive and path
$q	Equal symbol (=)
$t	System time
$v	DOS version number
$_	CR/LF (new line)
$$	Dollar sign symbol

If the dollar sign is followed by a character not found in the above table, the character pair in the prompt string is ignored.

For example, use the following PROMPT command to display the date and time in the prompt (omitting the year) and the current drive and directory:

```
PROMPT $d$h$h$h$h$h, $t$h$h$h$_$p$g
```

DOS may respond with a message similar to:

```
Not enough environment memory!
```

If this error message is displayed, you must shorten the PROMPT command parameter, increase the size of the MS-DOS environment or remove unnecessary variables from the environment.

Since escape characters can also be contained in the system prompt, it is possible to send escape sequences. This is useful only if the ANSI.SYS device driver is loaded through the CONFIG.SYS file. ANSI.SYS lets you use the standard control sequences. For example, the following PROMPT command displays the current directory in reverse video if ANSI.SYS is active:

```
PROMPT $e[7m$p$e[0m$g
```

☞ The <, >, = and | symbols are also used in normal DOS commands for operations like loading and saving files. This is the reason why they must be replaced by the escape sequences $g, $l and $b.

See also: PATH, SET

QBASIC

(external)

Starts the MS-DOS QBasic program.

Syntax: QBASIC [/b][/editor][/g][/h][/mbf][/nohi][/run] [*drive:*][*path*]*filename*

[*drive:*][*path*]*filename*
Specifies the name of an optional file for QBasic to load when the program begins.

Switches:

/b	Displays QBasic in black and white on a color monitor.
/editor	Calls the MS-DOS full screen text editor.
/g	Provides the fastest update for a CGA monitor.
/h	Displays the maximum number of lines possible for your screen.
/mbf	Converts the built-in function MKS$, MKD$, CVS and CVD to MKSMBF$, MKDMBF$, CVSMBF and CVDMBF.
/nohi	Permits using a monitor that doesn't support high-intensity video. This switch does not work with COMPAQ laptop computers.
/run	Loads and runs a specified BASIC program. (A filename must be specified.)

QBasic provides a complete programming environment for the BASIC language. It also includes extensive online help screens. For more information about using QBasic, refer to Chapter 16.

RD

(resident)

Removes a directory.

Syntax: RD [*drive:*]*path*

[*drive:*][*path*]
Specifies the location and name of the directory to be removed.

Description:

RD removes a subdirectory from its parent directory. The specified subdirectory cannot contain any files or subdirectories.

You cannot remove the current directory or the root directory.

For example, to delete subdirectory SAM from the directory C:\NAMES, type:

```
A>RD C:\NAMES\SAM
```

Or, if the current directory is NAMES, type:

```
C:\NAMES>RD SAM
```

☞ RD should not be used if one of the JOIN, ASSIGN or SUBST commands is active.

See also: RMDIR

RECOVER

(external)

Recovers readable information from a defective diskette.

Syntax: RECOVER [*drive:*][*path*]*filename*

To recover the files on a diskette when the directory is unusable, type:

RECOVER *drive*

NOTE: Do not attempt to recover all files in a directory unless it's absolutely necessary.

drive: pathname
The path and name of the file to be recovered. Wildcards are allowed, but only the first matching filename is rebuilt.

drive: The drive (in the second optional syntax) containing a damaged directory that will be recovered in its entirety.

Description:

RECOVER rebuilds files from a disk that contains bad sectors or rebuilds all files on a disk whose directory has been damaged. The data in the bad sectors is not recovered.

When a single file is RECOVERed, a copy of the file is made without the data in the bad sectors. You'll probably have to edit and delete erroneous parts of the file using a text editor or another method.

When a complete disk is RECOVERed, the directory structure is rebuilt using the File Allocation Table (FAT). The recovered files are renamed to FILEnnnn.REC where nnnn starts at 0001. You'll have to examine each of these files to determine their previous contents.

☞ Use the second form of RECOVER only as a last resort. Also, don't use RECOVER with a SHAREd drive, or with drives which have been redirected with JOIN, SUBST or ASSIGN.

REM

(resident)

Inserts a comment into a batch file or in CONFIG.SYS.

Syntax: REM *comment*

comment	Any string of characters to be included as a comment.

MS-DOS ignores any line in a batch file or CONFIG.SYS which begins with REM. If ECHO is ON, the line is displayed on the screen but any executable commands in the line are ignored.

IF ECHO is set to OFF, the comments are not displayed on the screen. Redirection or pipe symbols are not permitted with a batch file comment.

RENAME

(resident)

Changes the name of a file or group of files.

Syntax: RENAME [*drive:*][*path*]*oldname newname*
REN [*drive:*][*path*]*oldname newname*

drive: pathname oldname	The name of the file to be renamed. Wildcards are allowed.
newname	The new filename for *drive: pathname*.

Description:

RENAME or REN renames one or more files. The contents of the file remain unchanged.

If the two names are identical, if the file to be renamed does not exist or if filename *newname* already exists, the following error message is displayed:

```
Duplicate file name or file not found
```

RENAME displays the previous message and does not change the filename. Either delete one file or enter a different *newname*.

To change the name of the file READ.ME in the directory \COMPLEX\SCOMPLEX\ECOMPLEX to IGNORE.ME, use the following command:

```
REN \COMPLEX\SCOMPLEX\ECOMPLEX\READ.ME IGNORE.ME
```

To rename all files with the extension .TXT as files which start with the letter "T" and have an extension of .BAK, use the following command. Only the first letter of the renamed file and the filename extension will be replaced with "T":

```
REN *.TXT T*.BAK
```

See also: ATTRIB, COPY, DEL

REPLACE

(external)

Replaces files in the specified destination directory with files with the same name from the source directory.

Syntax: REPLACE [*drive1*:]*path1*]*filename* [*drive2*:][*path2*]
[/A][/p]{/r][/w]
REPLACE [*drive1*:]*path1*]*filename* [*drive2*:][*path2*]
[/p]{/r][/s][/w][/u]

drive1:path1 filename

Specifies the drive, path and name of the source file. Wildcards are allowed.

drive2: path2

Specifies the destination for the new files.

Switches:

/A	Add files to the destination. Only files that do not already exist are copied from the source. This switch cannot be used with /s or /u.
/p	Prompt before replacing a destination file or adding a source file.
/r	Replace read-only files as well as unprotected files in the destination.
/s	Search all subdirectories of the destination for matching source files. Switches /s and /A cannot be used together.
/w	Waits for a disk to be inserted before beginning to search for files.
/u	Updates only files in the destination directory that are older than those in the source directory. This switch cannot be used with /s or /a.

Description:

REPLACE selectively copies files from *drive1:path1filename* to the destination disk. This operation replaces the files on the destination disk with more recent versions of the files (updates files) or adds new files to the destination disk if they are not already there.

For example, the following command replaces all of the Pascal programs in directory \PASCAL on drive C: with Pascal programs of the same name on drive A:. DOS prompts you for confirmation before overwriting each file:

```
REPLACE A:\PASCOPY\*.PAS C:\PASCAL /s /p
```

When completed, REPLACE returns the following values to a batch file that can be tested with IF ERRORLEVEL:

Code	Meaning
0	Files added or replaced successfully
2	Source files not found
3	Source/destination path not found
5	User does not have access to the files to be replaced
8	Not enough memory
11	Syntax error in command line

See also: BACKUP, COPY, RESTORE, XCOPY

RESTORE

(external)

Restores files that were backed up using the BACKUP command.

Syntax: RESTORE *drive1*: [*drive2*:][*path filename*] [/s][/p][/b:*date*][/a:*date*][/e:*time*][/l:*time*][/m][/n][/d]

drive1:	Specifies the drive containing the backup files.
drive2:	Specifies the drive to which the files are to be restored. If omitted, the current drive is assumed.
path filename	Specifies the path and name of the file to be restored. Wildcards are allowed. If omitted, all files in the current directory of *drive2* are restored.

Switches:	/s	Restore all subdirectories.
	/p	Prompt before restoring read-only files or files that have changed since the last backup.
	/a:*date*	Restore only files that were modified on or after the specified *date*. The format for entering the date can vary according to the country setting in your CONFIG.SYS file.
	/b:*date*	Restore only files that were modified on or before the specified *date*. The format for entering the date can vary according to the country setting in your CONFIG.SYS file.
	/e:*time*	Restore only files that were modified before the specified *time*. The format for entering the time can vary according to the country setting in your CONFIG.SYS file.
	/l:*time*	Restore only files that were modified at or after the specified *time*. The format for entering the time can vary according to the country setting in your CONFIG.SYS file.
	/m	Restore only files modified since the last BACKUP.
	/n	Restore only files no longer on the destination. Existing files on the destination are not restored.
	/d	Display files on *drive1* that match specifications.

Description:

RESTORE copies files that were backed up by the BACKUP command. Only files backed up by BACKUP can be restored. The files are copied into the same directory from which they were backed up. You cannot RESTORE a file to a different directory.

For example, to restore all of the files on the diskette in drive B: that have been changed since the last backup (the diskette with the backup files is in drive A:), use the following:

```
RESTORE A: B: /m
```

The following command replaces all files with a .C extension, in the directory \CPROGS on drive A:, that have been changed since 11-5-1986. Read-only files are confirmed:

```
RESTORE A: C:\CPROGS\*.C /b:11-5-86 /a /p
```

When completed with its tasks, RESTORE returns the following values to a batch file that may be tested with IF ERRORLEVEL:

Code	Meaning
0	RESTORE successful
1	No files were found to restore
2	Some files could not be restored
3	RESTORE was terminated by the user with Ctrl+Pause (also referred to as the Break key)
4	Hardware error (*critical error*) terminated program

☞ Do not use RESTORE if a drive has been redirected with JOIN, SUBST or ASSIGN.

See also: BACKUP, COPY, DISKCOPY, REPLACE, XCOPY

RMDIR

(resident)

Deletes a directory.

Syntax: RMDIR [*drive:*]*path*
RD [*drive:*]*path*

drive:	The drive that contains the directory to be removed.
path	The path and/or name of the directory to be removed.

Description:

RMDIR or RD removes a subdirectory from its parent directory. The specified subdirectory cannot contain any files or subdirectories.

You cannot remove the current directory or the root directory.

For example, to delete subdirectory SAM from the directory C:\NAMES, type:

```
A>RD C:\NAMES\SAM
```

Or, if the current directory is NAMES, type:

```
C:\NAMES>RD SAM
```

☞ Neither RMDIR nor RD should be used if one of the JOIN, ASSIGN or SUBST commands is active.

See also: CD, CHDIR, MD, MKDIR

SET

(resident)

Displays, sets or removes environment variables.

Syntax: SET [*variable*=[*value*]]

variable	A variable name. All characters in *variable* are internally converted to uppercase characters.
value	The string of characters that specifies the current value of *variable*.

Description:

SET is used to define, display or remove an environment variable name and value.

If both *variable* and *value* are omitted, all of the current environment variable names and values are displayed.

IF *value* is omitted, *variable* and its *value* are deleted from the environment.

For example, one of the environment variables is called COMSPEC. It contains the path used to reload COMMAND.COM. Other programs can then reference COMSPEC as necessary.

Two other standard environment names are PATH and PROMPT (not to be confused with the DOS commands of the same name). These names can be displayed and modified with the SET command or the PATH and PROMPT commands.

The following command informs a program that the user is named Smith (provided the program reads the environment variable USER):

```
SET USER=Smith
```

The next command deletes the variable USER:

```
SET USER=
```

The following command tells DOS that COMMAND.COM is located in a different location:

```
SET comspec=C:\DOS\COMMAND.COM
```

Entering SET without parameters displays the current environment:

```
SET
COMSPEC=:\DOS\COMMAND.COM
PATH=C:\DOS
LIB=C:\LIB
TMP=C:\
USER=Smith
```

☞ If you load a new command processor and then modify the environment, the changes made will not be preserved following the EXIT command. The environment of the calling (original) command processor will be restored.

See also: APPEND, COMMAND, PATH, PROMPT

SETVER

(external)

Sets the version number that MS-DOS reports to a program.

Syntax: SETVER [*drive:*][*path*][*filename* n.nn]
SETVER [*drive:*][*path*][*filename*] [/delete][/quiet]

To display the current version table, type:

SETVER [*drive:*][*path*]

drive:path	Specifies the location of the SETVER.EXE file.
filename	The filename of the program (.EXE or .COM) to be added to the version table. Wildcards are not permitted.
n.nn	Specifies the MS-DOS version number (e.g., 3.3 or 4.01) for MS-DOS 5.0 to report to the specified program.

Switches: /delete

Deletes the version table for the specified program file.

/quiet

Does not display messages as entries are deleted from the version table.

Description:

SETVER is a device driver and must be installed from your CONFIG.SYS file with the following line:

```
DEVICE=SETVER.EXE
```

Many applications run only on a specific version of the MS-DOS operating system. The SETVER command is used to "trick" the application into thinking that MS-DOS 5.0 is really a different version.

As developers upgrade products to work with MS-DOS 5.0, SETVER will be used less frequently.

SETVER Exit Error Codes

0	SETVER completed successfully
1	Invalid command switch specified
2	Invalid filename specified
3	Insufficient memory
4	Invalid DOS version number specified
5	SETVER could not find the specified entry in the version table
6	SETVER.EXE not installed
7	Invalid drive specified
8	Too many command line parameters specified
9	Missing command line parameters
10	Error while reading SETVER.EXE file
11	SETVER.EXE has been corrupted
12	SETVER.EXE does not support version table
13	Insufficient space in the version table for new entry
14	Error while writing to SETVER.EXE

SHARE

(external)

Installs file sharing and locking capabilities on your hard drive.

Syntax: SHARE [/f:*file_buffer*][/l:*max_locks*]

You may also install SHARE.EXE from your CONFIG.SYS file by using the following command line:

INSTALL=[*drive:*][path] SHARE.EXE [/f:*file_buffer*] [/l:*max_locks*]

file_buffer	Specifies the number of bytes of memory to be reserved for storing file sharing information. The default size is 2048 bytes.
max_locks	The maximum number of files that may be locked at one time. The default value is 20 files.

Switches:

/l:*max_locks*
Specifies the number of files that can be locked.

/f:*file_buffer*
Sets memory space available for file sharing.

Description:

SHARE loads a program that allows files to be shared. If several PC's are used in a network, a method for sharing files must be implemented to prevent more than one user from accessing the same file at the same time.

Preventing other users from accessing a file on a network is known as *locking*. The /l switch specifies how many files can be locked. The system normally reserves space for 20 locked files.

The /f switch sets the memory space available for file sharing. Normally 2048 bytes are set up for this data. The system needs 11 bytes plus the length of the filename for each file.

The following command increases the size of both default values:

```
SHARE /f:4096 /l:40
```

SHIFT

(resident)

Changes the position of replaceable variables in a batch file.

Syntax: SHIFT

The SHIFT command changes the values of the replaceable parameters (%0 through %9) by copying each parameter into the previous parameter. For example, parameter %1 is copied into %0, %2 is copied into %1, %3 is copied into %2, etc.

Batch files can perform the same operation on any number of parameters.

SORT

(external)

Sorts data into ascending and descending order.

Syntax:

SORT [/r][/+n][<][*drive1:*][*path1*]*filename1*
[>[*drive2*:][*path2*]*filename2*][*command* |]
SORT [/r][/+n][>[*drive2*:][*path2*]*filename2*]

drive1:	Drive containing data to be sorted.
path1	Path to data to be sorted.
filename1	Input file with data to be sorted.
drive2:	Drive for output file for sorted data.
path2	Path to file for sorted data.
filename2	File for storing sorted data.

Switches:

/r	Sort the input data in descending (reverse) order: Z, Y, X, ... A.
/+n	Sort beginning at column n. For example, you can sort a disk directory according to size by beginning the sort at column 12.

Description:

SORT reorders the lines of the standard input file into ascending or descending sequence and rewrites them to the standard output file.

SORT is normally used by redirecting the standard input and output files using the < and > symbols. For example, the following sorts a file named INPUT and rewrites the sorted data to a file named OUTPUT:

```
SORT < INPUT > OUTPUT
```

You should use a different name for both files; otherwise the contents of the input file might be overwritten.

SORT does not distinguish between upper and lowercase characters. SORT can sort files up to 63K in size.

To enter a list of words from the keyboard and sort them, use the SORT command with no switches. Press Ctrl+Z following the last entry. SORT then displays the sorted word list on the screen.

The following command lets you enter data from the keyboard and outputs the sorted list to the printer:

```
SORT > PRN
```

You can chain SORT to another DOS command (using the pipe symbol) to send the output of the first command to the second command. The following chained commands sort the directory and send it to the printer:

```
DIR | SORT > PRN
```

See also: FIND, MORE

SUBST

(external)

Associates a path with a drive identifier.

Syntax:

SUBST [*drive1*: *drive2:path*]
SUBST *drive1*: /d

drive1:	The drive identifier to be substituted for a subdirectory.
drive2:path	The drive and path to be accessed. *drive1:* and *drive2:* must be different.

Switches: /d Disable the redirection of the specified drive. The data on this drive can be accessed again or the drive can be redirected to a new directory.

Description:

SUBST redirects all references from a drive to a directory.

Some applications, such as Wordstar, do not support directory structures. You can use the SUBST command to overcome this limitation and access files in a directory as if the directory were a disk drive. For example, the following allows Wordstar to access text files in the \WSFILES subdirectory of drive C: as if they were located on drive B:

```
SUBST B: C:\WSFILES
```

SUBST can also be used to shorten long pathnames. For example, you can define drive D: as the drive on which all DOS commands are located. Then you can execute all of the DOS commands by typing D: before the command:

```
A> SUBST D: C:\DOS
A>D:ATTRIB *.EXE
```

Omit all parameters to display the current list of substitutions.

You can use the TRUENAME command to display the real name (the original directory name) of the pseudo drive.

To do this, simply type the command along with the drive whose true name you want to display:

```
TRUENAME drive
```

The screen then displays the correct name of the directory.

See also: ASSIGN, JOIN

SYS

(external)

Copies MS-DOS system files and command interpreter to a specified disk.

Syntax: SYS [*drive1:*][*path*] *drive2:*

drive1: path	Location of the system files. If not specified, MS-DOS searches the root directory.
drive2:	The destination disk to which the system files are to be copied.

Description:

SYS transfers the two system files to the disk in the specified drive. SYS also transfers the COMMAND.COM file to the disk.

The disk in *drive:* must be empty or must have been previously formatted with the /s switch of the FORMAT command. This ensures that the first two entries in the disk's directory are reserved for the two system files.

TIME

(resident)

Displays or sets system time.

Syntax: TIME [*time*]

time The new time in the format defined by the country setting in your CONFIG.SYS files. Allowable values for hours, minutes and seconds are:

HH	Hours (0-23)
MM	Minutes (0-59)
SS	Seconds (0-59) *Optional*
TT	Tenths of seconds (0-99) *Optional*

Description:

TIME sets the system clock.

If *time* is omitted, the current system time is displayed and you are prompted to enter a new time.

See also: DATE

TREE

(external)

Graphically displays the directory structure of a drive or path.

Syntax: TREE [*drive:*][*path*][/f][a]

drive:path

The drive and path containing the directories to be displayed. If omitted, the current drive is assumed.

Switches:

/f	Display filenames within each directory.
/a	Specifies that the tree structure should display text characters instead of graphics characters.

Description:

TREE displays the structure of the directories of the disk in the specified drive.

Type the following command to output the directory structure to your printer:

```
TREE C: /f/a >PRN
```

In order to display all subdirectories on the current disk and drive, type:

```
TREE \
```

TYPE

(resident)

Displays the contents of a text file.

Syntax: TYPE [*drive:*][*pathname*]*filename*

drive: pathname	The path and drive of the file to be displayed.
filename	The name of the file to be displayed. Wildcards are not allowed.

Description:

TYPE displays the contents of a file to the standard output device.

If the file contains a tab (code 09h), it is replaced by spaces until the next tab position (every 8 columns).

You can use TYPE with MORE to prevent the displayed output from scrolling off the screen. So, the command:

```
TYPE read.me | MORE
```

is the same as:

```
MORE < read.me
```

You can also redirect the output to your printer:

```
TYPE read.me > prn
```

UNDELETE

(external)

Recovers file previously deleted with the DEL command.

Syntax: UNDELETE [[*drive:*][*path*]*filename*][/list | /all][/dos | /dt]

drive:path filename

The location of the file or files to be recovered.

Switches:

/list	Lists all deleted files which may be recovered, but does not recover any files.
/all	Recovers all deleted files without prompting for confirmation.
/dos	Recovers only those files that are internally listed as deleted by MS-DOS.
/dt	Recovers only files listed in the MIRROR command's deletion tracking file. Before recovering each file, a confirmation prompt is required.

UNDELETE recovers accidentally deleted files whenever possible.

UNDELETE cannot recover a directory that has been removed and cannot recover a file if the directory which contained the file has been removed. The UNFORMAT command may be able to assist you with recovering such files.

See also: MIRROR

UNFORMAT

(external)

Restores a disk erased by the FORMAT command or restructured by the RECOVER command.

Syntax:
UNFORMAT *drive:* [/j]
UNFORMAT *drive*: [/u][/l][/test][/p]
UNFORMAT [/partn][/l]

drive:	Specifies the drive that contains the disk to be recovered.

Switches:

/j	Verifies that a file created by MIRROR has been saved and that it agrees with the system information on the disk. No other switch may be used with this switch.
/u	Unformats a disk without using the MIRROR file.
/l	Lists every file and subdirectory when used without the /PARTN switch. When used with /PARTN, UNFORMAT displays the partition table of the current drive.
/test	Simulates how UNFORMAT would restore the information on the disk without actually performing UNFORMAT.
/p	Sends output to the printer on LPT1:.
/partn	Restores a corrupted partition table of a hard drive. This switch requires PARTNSAV.FIL, which was created by using MIRROR with the /partn switch.

Although UNFORMAT is a powerful command, it has some limitations. If a disk was formatted with the /u (unconditional format) switch of the FORMAT command, it cannot be recovered.

When used in conjunction with MIRROR, UNFORMAT can usually restore an accidentally formatted disk.

See also: MIRROR

VER

(resident)

Displays the version of MS-DOS being run.

Syntax: VER

Description:

VER displays the version number of DOS. Many applications test the DOS version number to ensure that the application is compatible with the operating system.

For example:

```
C>VER
MS-DOS Version 5.00
C>
```

See also: SETVER

VERIFY

(resident)

Tells MS-DOS whether to verify that files are written correctly to disk.

Syntax: VERIFY [on | off]

on	Enables write verification.
off	Disables write verification.

Description:

VERIFY ensures that data written to a disk is verified.

If ON or OFF is omitted, the current status of write verification is displayed:

```
C>VERIFY
VERIFY is off
```

VOL

(resident)

Displays the volume label (if any) and volume serial number of a disk.

Syntax: VOL [*drive:*]

drive:	The drive containing the disk whose volume label is displayed. If omitted, the current drive is assumed.

Description:

VOL displays the volume label of a disk. For example:

```
A>VOL C:
Volume in drive C is PROGRAMS
Volume Serial Number is 1699-5914
```

If a disk was not labeled during formatting or with the LABEL command, the following message is displayed:

```
Volume in drive A has no label
```

XCOPY

(external)

Copies files (except hidden and system files) and directories, including subdirectories.

Syntax: XCOPY [*sourcedrive*:][*sourcepath*]*filename1*
[*dest_drive*:][*dest_path*]*filename2*][/a |
/m][/d:*date*][/p][/s][/e][/v][/w]

sourcedrive: *sourcepath*	The drive and path from which the source file will be copied.
filename1	The name of the file to be copied. Wildcards are allowed. If *drive*: is omitted, the current drive is assumed. If only *drive*: is specified, all of the files in the current directory are copied. If you specify only a path without a filename, all files in the specified directory are copied.
dest_drive:*dest_path*	The destination drive and path.
filename2	Specifies the name of the destination file. Wildcards are allowed. If omitted, the current drive and directory are assumed.

Switches:

/a	Copy files with set archive flags. The flag isn't changed.
/m	Copy only files with set archive flags. This switch resets (turns off) the archive flags.
/d:*date*	Copy only files created or modified on or after the specified date.
/p	Prompts before copying each file to the destination.
/s	Copy directories and subdirectories.
/e	Copy subdirectories in the destination path, even if they are empty. Used in conjunction with /s.
/v	Verify the copied files.
/w	Display a message and wait for a keypress before starting to copy files. This switch allows switching diskettes.

Description:

XCOPY makes a copy of a file. XCOPY can also copy existing directories and subdirectories if specified by the /e and /s switches. For example, the following use of XCOPY copies all files, directories and subdirectories from drive A: to drive B:, then verifies the copied files:

```
XCOPY A:*.* B: /e /s /v
```

☞ Do not use XCOPY when any of the ASSIGN, JOIN or SUBST commands are active.

See also: BACKUP, COPY, DISKCOPY, REPLACE, RESTORE

Appendix A: Device Drivers

A device driver is a program used to control a peripheral device.

Normally each device (such as a disk drive, system clock or keyboard) needs a device driver. The standard device drivers are automatically loaded by DOS. You can also install device drivers that replace the standard input and output devices, or create a virtual disk (also known as a RAM disk).

The syntax for installing a device driver from your CONFIG.SYS file is:

```
DEVICE=[drive:]path driver
```

The following commands install a device driver:

```
DEVICE=ANSI.SYS
DEVICE=DISPLAY.SYS
DEVICE=DRIVER.SYS
DEVICE=EGA.SYS
DEVICE=HIMEM.SYS
DEVICE=KEYBOARD.SYS
DEVICE=PRINTER.SYS
DEVICE=RAMDRIVE.SYS
DEVICE=SMARTDRV.SYS
```

ANSI.SYS

Driver which supports use of ANSI escape sequences.

Syntax: DEVICE= [*drive:*][*path*]ANSI.SYS [/x][/k]

drive: path Drive and path for ANSI.SYS file.

Switches:

/x	Remap extended keys independently on 101-key keyboards.
/k	Ignore extended keys on 101-key keyboards.

The ANSI.SYS driver enables the ANSI escape sequences for screen control. These escape sequences perform operations such as clearing the screen and defining the cursor position.

Escape sequences begin with the hexadecimal codes 01Ah and 05Bh (^Z[). These characters may be entered by pressing Alt+2 6, Alt+9 1.

The following is the list of recognized escape sequences. Parameters are shown in *italics*:

ESC[2J	Erase screen and move the cursor to the Home position (line 0, column 0).
ESC[K	Erase all characters from the cursor position (including the character under the cursor) to end of line.
ESC[x;yH	Define the cursor position. If no position is specified, cursor is moved to home position.
ESC[x;yf	Positions the cursor in column *x*, line *y*. If one or both of the parameters are omitted, the default is 1.
ESC[xA	Move cursor up. Moves the cursor up *x* lines without changing the column. If the *x* parameter is omitted, the cursor is moved one line. The cursor cannot be moved past the top of the screen.
ESC[xB	Move cursor down. Moves the cursor down *x* lines without changing the column. If the *x* parameter is omitted, the cursor is moved one line. The cursor cannot be moved past the bottom of the screen.
ESC[xC	Move cursor right. Moves the cursor *x* characters to the right. If the *x* parameter is omitted, the cursor is moved one character. The line remains unchanged. The cursor cannot be moved past the rightmost column.

ESC[xD Move cursor left. Moves the cursor *x* characters to the left. If the *x* parameter is omitted, the cursor is moved one character. The line remains unchanged. The cursor cannot be moved past the leftmost column.

ESC[s Save current cursor position.

ESC[u Restore cursor to saved position.

ESC[=xh Set screen mode.

0	40x25 monochrome (text)
1	40x25 color (text)
2	80x25 monochrome (text)
3	80x25 color (text)
4	320x200 4-color (graphics)
5	320x200 monochrome (graphics)
6	640x200 monochrome (graphics)
7	Enable word wrap
13	320x200 color (graphics)
14	640x200 color (16 color graphics)
15	640x350 (monochrome (2-color graphics)
16	640x350 color (16 color graphics)
17	640x480 monochrome (2 color graphics)
18	640x480 color (16 color graphics)
19	320x200 color (256 color graphics)

ESC[x;y;...;zm

Set screen attribute. The following control sequences can be accessed in normal text mode, from most video controller cards:

Text Attributes

0	None
1	Bold
2	Normal
3	Italic
5	Blinking
6	Fast blinking
7	Reverse video
8	Invisible

Foreground Colors

30	Black foreground
31	Red foreground
32	Green foreground
33	Yellow foreground
34	Blue foreground
35	Magenta foreground
36	Cyan foreground
37	White foreground

Background Colors

40	Black background
41	Red background
42	Green background
43	Yellow background
44	Blue background
45	Magenta background
46	Cyan background
47	White background

Parameters 30 through 47 satisfy the requirements for the ISO 6429 standard.

ESC[x;y;...p

Assign string y+...+z to key x

Allows you to assign string variables to a specific key. For example, you could assign the DOS commands to their starting letters plus the Alt key. The string must be enclosed in quotes, or you can define the ASCII code or a single character in decimal. The parameters are separated by semicolons.

You can define more than one key in a single escape sequence by repeating Esc-[. A semicolon (;) cannot precede the terminating p.

All of the codes listed in the Key Code table (on the next page) are preceded by a zero (0). For example, the following causes Alt+D to clear the screen and display the current directory:

```
PROMPT $e[0;32;"cls";13;"dir";13p

PROMPT
```

A null PROMPT command restores the standard system prompt, and reassign the default key definitions. To restore Alt+D to its normal state, type the following:

```
PROMPT $e[0;32;0;13p
```

Key code 13 represents the Enter key, and is needed to end a line. Key code 0;32 represents the key combination Alt+D (see the Key Code Table below).

☞ Some applications may not operate correctly with a redefined keyboard.

Key Code Table

Code	Key	Code	Key	Code	Key
3	Null	63	F5	101	Ctrl+F8
13	Enter	64	F6	102	Ctrl+F9
15	Shift+Tab	65	F7	103	Ctrl+F10
16	Alt+Q	66	F8	104	Alt+F1
17	Alt+W	68	F9	105	Alt+F2
18	Alt+E	69	F10	106	Alt+F3
19	Alt+R	71	Home	107	Alt+F4
20	Alt+T	72	↑	108	Alt+F5
21	Alt+Y	73	PgUp	109	Alt+F6
22	Alt+U	75	←	110	Alt+F7
23	Alt+I	77	→	111	Alt+F8
24	Alt+O	79	End	112	Alt+F9
25	Alt+P	80	↓	113	Alt+F10
30	Alt+A	81	PgDn	114	Ctrl+Prt Sc
31	Alt+S	82	Ins	115	Ctrl+←
32	Alt+D	83	Del	116	Ctrl+→
33	Alt+F	84	Shift+F1	117	Ctrl+End
34	Alt+G	85	Shift+F2	118	Ctrl+PgDn
35	Alt+H	86	Shift+F3	119	Ctrl+Home
36	Alt+J	87	Shift+F4	120	Alt+1
37	Alt+K	88	Shift+F5	121	Alt+2
38	Alt+L	89	Shift+F6	122	Alt+3
44	Alt+Z	90	Shift+F7	123	Alt+4
45	Alt+X	91	Shift+F8	124	Alt+5
46	Alt+C	92	Shift+F9	125	Alt+6
47	Alt+V	93	Shift+F10	126	Alt+7
48	Alt+B	94	Ctrl+F1	127	Alt+8
49	Alt+N	95	Ctrl+F2	128	Alt+9
50	Alt+M	96	Ctrl+F3	129	Alt+0
59	F1	97	Ctrl+F4	130	Alt+-
60	F2	98	Ctrl+F5	131	Alt+=
61	F3	99	Ctrl+F6	132	Ctrl+PgUp
62	F4	100	Ctrl+F7		

DISPLAY.SYS

Driver to support code page switching. Prepares screen for character set tables.

Syntax: DEVICE=[*drive:*][*path*]DISPLAY.SYS CON[:]=(*type*
[,[*hwcp*][,*n*]])
DEVICE=[*drive:*][*path*]DISPLAY.SYS CON[:]=(*type*
,[*hwcp*][,(*n*,*m*)]])

[*drive:*][*path*]	Defines the location of the DISPLAY.SYS file.
type	Specifies the display adapter type being used. Valid values are EGA and LCD. EGA supports both EGA and VGA video adapters. If *type* is omitted, DISPLAY.SYS checks the hardware to determine which display adapter is in use. CGA and MONO may also be specified, but these values have no effect.
hwcp	Defines the number of the code page supported. Valid values are 437, 850, 860, 863 and 865.
n	Number of code pages that can be supported.
m	Number of subfonts for each code page.

DISPLAY.SYS configures the screen device driver for use with the specified character set table.

Using DISPLAY.SYS with monochrome or CGA graphics cards has no effect as code page switching is not supported.

DRIVER.SYS

Installs a logical disk drive with different default characteristics from the physical disk drive.

Syntax: DEVICE=[*drive:*][*path*]DRIVER.SYS /d:*n*[/c][/f:*factor*][/h:*heads*][/s:*sectors*][/t:*tracks*]

[*drive:*][*path*]

Defines the location of the DRIVER.SYS file.

/d:*n*

Specifies the number of the physical disk drive. Valid numbers are 0 through 127.

/c

Specifies that the physical disk drive can detect whether the drive door is open or closed.

/f:*factor*

Specifies the type of disk drive.

0	160K/180K or 320K/360K
1	1.2 Meg
2	720K (3.5 inch disk)
7	1.44 Meg (3.5 inch disk)
9	2.88 Meg (3.5 inch disk)

The default value is 2. If the /f switch is used, the /h, /s and /t switches may be omitted.

/h:*heads*

Defines the number of heads in the disk drive. Valid values range from 1 through 99. The default value is 2. Refer to your drive manufacturers documentation to determine the correct value for your drive.

/s:*sectors*

Specifies the number of sectors per track. Valid values are 1 through 99. The default value is dependant on the value of /F:*factor* as shown in this table:

/f:0	/s:9
/f:1	/s:15
/f:2	/s:9
/f:7	/s:18
/f:9	/s:36

/t:*tracks*

Specifies the number of tracks per side. Valid values range from 1 through 999. The default value is 80, unless the /f switch is set to 0. In this case the default value is 40.

DRIVER.SYS cannot be used with hard drives.

EGA.SYS

Saves and restores the display when the Task Swapper is used with EGA monitors. (If you use an EGA monitor, EGA.SYS must be installed prior to using the Task Swapper from the MS-DOS Shell.)

Syntax: DEVICE=[*drive:*][*path*]EGA.SYS

[*drive:*][*path*]	Defines the location of the EGA.SYS file.

EMM386.EXE

Driver which simulates expanded memory using extended memory. Provides access to the upper memory area on computers using an 80386 (or higher) microprocessor.

Syntax: DEVICE=[*drive:*][*path*]EMM386.EXE [on | off | auto][*memory*][w=on | w=off][m*x* | frame=*address* | */pmmmm*][*pn=address*] [x=*mmmm-nnnn*] [i=*mmmm-nnnn*][b=*address*] [L=*minXMS*][a=*altregs*][h=*handles*][d=*nnn*][ram][noems]

[*drive:*][*path*]
Defines the location of the EMM386.EXE file.

[on | off | auto]
Set to on the device driver is activated. Set to off suspends device driver. Auto enables expanded memory support only when a program calls for it.

memory Specifies the amount of memory (in K) to allocate to EMM386.EXE. Valid values range from 16 to 32768. The default value is 256.

Switches:

w=on|off Enables or disables support for the Weitek coprocessor. The default setting is off.

m*x* Specifies the address of the page frame valid values are 1 through 14 as shown in the following table of base addresses:

1	C000h
2	C400h
3	C800h
4	CC00h
5	D000h
6	D400h
7	D800h
8	DC00h
9	E000h
10	8000h
11	8400h
12	8800h
13	8C00h
14	9000h

Values of 10 through 14 should only be used on computers with 512K of memory.

frame=*address*	Specifies the page frame segment base. Valid values for *address* are 8000h through 9000h and C000h through E000h, in increments of 400h.
/p*mmmm*	The address of the page frame. Valid values are 8000h through 9000h and C000h through E000h, in increments of 400h.
p*n*=*address*	Segment address of a specific page. *n* is the number of the page, *address* is the segment address. Valid values for *n* are 0 through 255. Valid values for *address* are 8000h through 9000h and C000h through E000h, in increments of 400h.
x=*mmmm-nnnn*	Prevents EMM386.EXE from using a specified range of addresses.
i=*mmmm-nnnn*	Specifies a range of address to be used for an EMS page.
b=*address*	Lowest segment address available for EMS page swapping.
L=*minXMS*	Ensures that a specified amount of extended memory is still available after loading EMM386.EXE. The default value is 0.
a=*altregs*	This switch is used for multitasking and specifies how many fast alternate register sets are to be allocated to EMM386.EXE.
h=*handles*	Specifies how many handles EMM386.EXE can use. Valid values are 2 through 255. The default value is 64.
d=*nnn*	Reserves memory for direct buffer access.
ram	Provides access to both the expanded memory area and the upper memory area.
noems	Provides access to the upper memory area but prevents accessing expanded memory.

The EMM386.EXE device driver uses extended memory to simulate expanded memory for programs that can use expanded memory. This device drive also enables loading programs and device drivers into the upper memory area.

HIMEM.SYS must be installed before using the DEVICE=EMM386.EXE command.

HIMEM.SYS

Driver which manages the use of extended memory.

Syntax: DEVICE=[*drive:*][*path*]HIMEM.SYS [/hmamin=*m*][/numhandles=*n*][/int15=*xxxx*][/machine: *xxx*][/a20control:ON|OFF][/shadowram ON|OFF][/cpuclock: ON|OFF]

[*drive:*][*path*]

Defines the location of the HIMEM.SYS file.

Switches: /hmamin=*m*

Defines the amount of memory (in kilobytes) a program must use before using the high memory area. Valid values are 0 through 63. The default value is 0.

/numhandles=*n*

Maximum number of extended memory block handles that can be used at the same time. Valid values are 1 through 128. The default setting is 32.

/int15=*xxxx*

Allocates a specified amount of memory for the Interrupt 15h interface.

/machine:*xxxx*

Specifies the a20 handler to be used. The default value is at or 1.

Code	Number	a20 handler
at	1	IBM PC/AT
ps2	2	IBM PS/2
pt1cascade	3	Phoenix Cascade BIOS
hpvectra	4	HP Vectra (A and A+)
att6300plus	5	AT&T 6300 Plus
acer1100	6	Acer 1100
toshiba	7	Toshiba 1600 and 1200XE
wyse	8	Wyse 12.5 MHz 286
tulip	9	Tulip SX
zenith	10	Zenith ZBIOS
at1	11	IBM PC/AT
at2	12	IBM PC/AT (alternative delay)
css	12	CSS Labs
at3	13	IBM PC/AT (alternative delay)
philips	13	Philips
fasthp	14	HP Vectra

/a20control:ON|OFF

Determines if HIMEM.SYS is to take control of the a20 line.

/shadowram ON|OFF

Determines whether HIMEM.SYS should switch off shadow RAM and add that RAM to its memory.

/cpuclock: ON|OFF

Determines if HIMEM.SYS is to affect the CPU clock.

HIMEM.SYS manages the way a program uses extended memory and the high memory area.

PRINTER.SYS

Driver which supports code page switching for the parallel ports PRN, LPT1, LPT2 and LPT3.

Syntax: DEVICE=[*drive:*][*path*]PRINTER.SYS
lpt*x*=(*type*[,[*hwcp*][,*n*]])

[*drive:*][*path*]
Defines the location of the PRINTER.SYS file.

Switches: lpt*x*
Defines the number of the parallel port for which code page support is required.

type
Defines the printer in use. Valid values for *type* are shown in the following table:

4201	IBM Proprinter Model 4201 family
	IBM Proprinter XL Model 4202
4208	IBM Proprinter X24 Model 4207
	IBM Proprinter XL24E Model 4208
5202	IBM Quietwriter III Model 5202

hwcp Specifies which code page is supported by your hardware.
n Defines the number of code pages supported.

PRINTER selects a code page and character set table for IBM Proprinter models 4201, 4202, 4207, 420 and IBM Quietwriter III model 5202. This command has no effect on other printers.

The following commands assign code page 437 and character table 1 to an IBM Model 4201 Proprinter:

```
DEVICE=PRINTER.SYS lpt1:=(4201,437,1)
```

RAMDRIVE.SYS

Driver to create a virtual disk.

Syntax: DEVICE=[*drive:*][*path*]RAMDRIVE.SYS
[*size*][*sectors*][/e|/a]
DEVICE=[*drive:*][*path*]RAMDRIVE.SYS
[*size*][*sectors*][*entries*][/e|/a]

[*drive:*][*path*]	Defines the location of the RAMDRIVE.SYS file.
size	Number of kilobytes to be reserved for the RAM disk. This value can range from 1 to the upper limit of memory. At least 64K of memory must be left available for applications.
sectors	Size of the virtual sector. *sectors* can be 128, 256 or 512 bytes.
entries	Number of virtual directory entries (from 2 to 512). The default is 64.

Switches:	/e	Specifies that the virtual disk is located in extended memory above the 1 Meg boundary.
	/a	Creates the RAM disk in expanded memory instead of extended or conventional memory.

A virtual disk (or RAM disk) is a device that simulates a disk drive. The virtual disk is actually located in the PC's memory.

The following example defines a RAM disk with a capacity of 150K. Each sector is 256 bytes long and 100 directory entries are allowed:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 150 256 100
```

SMARTDRV.SYS

Driver to create a disk cache in extended or expanded memory.

Syntax: DEVICE=[*drive:*][*path*]SMARTDRV.SYS [*InitSize*][*MinSize*][/a]

[*drive:*][*path*]	Specifies the location of the SMARTDRV.SYS file.
InitSize	Initial size (in kilobytes) of the cache. Valid sizes are 128 through 8192. The default value is 256.
MinSize	Defines the minimum cache size (in kilobytes).

Switches: /a Places the disk cache in expanded memory. If this switch is omitted, extended memory is used.

In order to use SMARTDRV.SYS, HIMEM.SYS (or another extended memory manger that conforms to the LIM/AST extended memory specification) must be installed first.

Do not run disk compression (defragmentation) programs while using SMARTDRV.SYS, unless the disk compression program is compatible with SMARTDRV.SYS. The BeckerTools Compress disk optimization program from Abacus is compatible with SMARTDRV.SYS

Appendix B: A Short History Of MS-DOS

MS-DOS's life began in the fall of 1981 with the invention of the first IBM PC. This PC had from 16 to 64K of main memory and used single sided diskettes with 160K storage capacity. MS-DOS 1.0, supplied with these PCs, supported equipment that would be spartan by today's standards and had fewer commands and options than the latest versions of MS-DOS.

Shortly thereafter, MS-DOS Version 1.1 appeared. It was able to support 320K of memory and also increased the storage capacity of diskettes by using both sides - up to 320K.

In 1983 the IBM PC/XT (XT is an abbreviation for eXTended), with a 10 Meg hard drive, appeared on the market. This hard drive could store 10 million characters. To manage the larger number of files, developers built into MS-DOS the option of setting up subdirectories. That is, there wasn't just one directory, but subdirectories where users could keep information that belonged together, just like the drawers of a desk. This new version of MS-DOS was Version 2.0. MS-DOS 2.0 made it possible for users to specify a search path for MS-DOS commands and programs with the PATH command.

In addition, Version 2.0 was better able to support new devices and peripherals. The new version had new commands for configuring the computer (DEVICE for adding additional utility programs, BUFFERS and FILES for adapting to different storage media) and a special driver called ANSI.SYS for extending the screen and keyboard control system.

In 1984 developers added the option of customizing for different countries. To do this, Version 2.1 got a program called COUNTRY.SYS. In the same year MS-DOS was translated into many different languages and licensed to many personal computer manufacturers. Thanks to this, Version 2.11 became widely distributed.

Next came the development of the successor to the PC/XT, the new PC/AT (referred to as AT nowadays). This new computer was not only faster, but could also manage more than 640K of memory and was outfitted with a hard drive and a disk drive with 1.2 Meg capacity. In 1984, MS-DOS 3.0 came out as a temporary solution for use with the AT, with support for the new disk drives and hard drives. In the same year, Version 3.1 came out. It offered network capabilities, simulated drives and keyboard adaptation to special characters such as accents in foreign countries.

With MS-DOS 3.1, it was possible for the first time to use extended memory for the AT. MS-DOS 3.1 included the driver VDISK.SYS, which could use a part of the main memory or extended memory for file storage (RAM disk).

In 1986 Version 3.2 brought support for the new 3.5 inch disk drives with 720K storage capacity. The configuration command, DRIVPARM, allowed users to change the parameters of drives. The new APPEND command not only made it possible to find commands in any directory (see PATH), but also allowed users to define a search path for files. MS-DOS finally had a command for copying files that included subdirectories and had special options for data backup: XCOPY.

In 1987 Version 3.3 appeared, still the most widespread version of MS-DOS. It allowed users to adapt to country specific characters that could be displayed on the screen and printed on IBM printers. You could also use CALL to call other batch files from a batch file as subroutines. What is more, the new configuration command, FASTOPEN, enabled users to find subdirectories and files quickly, speeding up their work with storage media.

Many believed that MS-DOS's successor, the multitasking operating system OS/2, meant the end of MS-DOS. Soon, however, it turned out that OS/2 required a lot of memory (from 2 Meg and upwards) and a speedy computer. Since memory was rather expensive at the time (1 Meg was about 500 dollars - today it's under 200 dollars), in 1988 another version appeared: MS-DOS 4.0. The most important innovations were:

- For the first time, a graphical user interface was available with the DOS Shell, which could be operated using pull-down menus, a mouse or a keyboard.

- The 32 Meg limit for hard drives was lifted. Hard drives could now be up to 2 gigabytes (2048 Meg) in size.

- EMM386.SYS and XMA2EMS.SYS enable users to manage expanded memory; some MS-DOS commands could also make use of it.

- Installation went smoothly, was menu-driven, the system files AUTOEXEC.BAT and CONFIG.SYS were automatically created during installation.

Finally: MS-DOS 5.0

For the first time there is a new version of MS-DOS that can not only do more on most computers, but still makes more memory available to applications. We'll tell you how this is possible and about all of the other innovations in the next section.

Hassle-free installation

Installation is not only menu driven and extremely simple, MS-DOS 5.0 also automatically backs up all of the important data of the hard drive onto a backup diskette and can restore the status prior to installation if problems develop

(UNINSTALL). Also, existing versions of DOS are not overwritten, but backed up to a special directory. This guarantees a maximum amount of security. By the way, you don't even have to boot MS-DOS 5.0 for the installation. On the contrary, you can start the installation program, SETUP, from other versions of DOS, too.

A DOS Shell for everyone

If you set up the MS-DOS Shell during installation, the next time you start the system you will automatically go to its user interface. The elements of the MS-DOS Shell borrow heavily from the elements in Windows 3.0 (Program Manager, File Manager, Task Manager, etc.).

The new MS-DOS Shell operates according to SAA (System Application Architecture) standards. You can press F10 to activate the menu, or use Alt key combinations.

The new MS-DOS Shell has a lot of other improvements: Along with colors, users can set different graphics modes. It is also possible to divide up different tasks practically, since you can finally create new groups in program groups. This allows you to create specific user shells.

"Multitasking" in the DOS Shell

Although MS-DOS 5.0 remains a single-user, single-program operating system, the Task Swapper in the MS-DOS Shell can easily overcome this limitation of running one program at a time. If you activate the appropriate menu item in the MS-DOS Shell, the Program List opens one area for active programs. When you then start a program from the MS-DOS Shell, you can press Ctrl + Alt to switch to the next program, and press Ctrl + Esc to return to the MS-DOS Shell. The current program is "frozen" and saved on the hard drive.

In the MS-DOS Shell, the program appears in the list of active programs. In this way you can start several programs and switch between them. Although you cannot run several programs parallel to each other the way you can in real multitasking, switching between several programs is enough in most cases.

New QBasic

The QBasic interpreter comes with MS-DOS 5.0, while earlier versions up to MS-DOS 4.01 had GW-BASIC. With QBasic you can develop programs comfortably, as the editor also supports a mouse. Anyone who is used to BASIC, a powerful, user friendly programming language, and would like to create EXE files, can just use QuickBASIC 4.5 to compile QBasic programs.

New Full Screen Editor

Users who had a hard time getting used to EDLIN, the line oriented editor that came with older versions of MS-DOS, can breathe a sigh of relief with MS-DOS 5.0: The MS-DOS Editor (EDIT.COM) which comes with MS-DOS 5.0 is very similar to the editor of QBasic. You might look at the file length of EDIT.COM (less than 1K) and wonder how Microsoft was able to pack such a versatile editor in such a small program. Call QBasic with /Editor:. That should reveal the secret, because the small program, EDIT.COM does all of that.

Help included

The MS-DOS commands come with an integrated help screen, which you can activate with the /? switch. This help screen is not only available for the external commands (as is the case with the competition, DR DOS 5.0), but also for the resident commands stored in the command interpreter. That means that you don't have to go through reference manuals looking up the exact syntax of a command. The HELP command displays a list of the commands for which MS-DOS 5.0 provides a help screen.

New commands and extensions

MS-DOS 5.0 contains a number of extensions and improvements. For example, the DIR command can sort files according to various criteria (name, date, file size) and can also work over subdirectories. Aside from that, users can limit the display to files with certain file attributes. The ATTRIB command also allows users to change the attributes "System" and "Hidden".

A series of new commands expands MS-DOS 5.0's performance. The UNFORMAT command allows you to restore a hard drive or diskette that was accidentally formatted to its original state. You can use UNDELETE to recover deleted files. With SETVER you can run individual programs under different versions of MS-DOS. SETVER manages a list which is stored in the MSDOS.SYS file, making changes the next time you boot the computer. A number of programs are already included in this version list and can be displayed by entering SETVER without any parameters.

The new DOSKEY command lets users reactivate command lines entered previously and edit them comfortably with the cursor keys. One error in a long command line no longer leads to frustration. What's more, DOSKEY has a type of "built in batch language" with which users can assign extensive command sequences to a single command. Microsoft refers to these command sequences as macros. These macros are very similar to batch files—you can even pass parameters.

The big advantage compared to other utility programs that are designed for the same purpose: If you assign the name of an internal MS-DOS command such as DIR to one of these macros, the macro is called first, superseding the original command for the user's own command. It allows you to create your own "personal" MS-DOS.

MS-DOS 5.0 and memory requirements

Users who are excited about the many new options and improvements but are afraid that the new DOS could require more memory, leaving even less memory available for applications, can breathe a sigh of relief if they own an AT with at least 1 Meg of main memory. MS-DOS, with the help of the extended memory manager HIMEM.SYS, uses the first 64K of extended memory (High Memory), placing 50K more of main memory at your disposal for applications. If you start MS-DOS without any driver program (in a sense, naked), this leaves about 623K of memory available for applications.

To make use of this option, you need at least an AT with extended memory and the appropriate commands inserted in your CONFIG.SYS file. When you install MS-DOS 5.0, these lines are automatically added to CONFIG.SYS.

By the way, all the important MS-DOS commands prompt you for the MS-DOS version you are using (i.e., you couldn't boot from MS-DOS 3.3 and then begin using MS-DOS 4.0 commands). Other than that, though, MS-DOS is upwardly compatible. You can install a new version of MS-DOS and continue working with all of the programs and data.

Special features on disk drives

Since we based all of our explanations in this book on the assumption that you have a PC with a hard drive, we want to give you a few tips about how to use this book if you own a PC that has two disk drives and no hard drive.

Installation

You also use SETUP to install MS-DOS 5.0 on a system without a hard drive, because the programs and files of the diskettes have to be unpacked. The installation program determines that there is no hard drive and shows the necessary number and labels of the diskettes for the installation.

✎ If for some reason SETUP still thinks the system has a hard drive, you can use the special /F switch to force an installation on diskettes.

Working with MS-DOS 5.0

To be able to work with your computer, you have to start it each time with the STARTUP diskette, created during installation, in drive A:.

MS-DOS is distributed over a number of diskettes so that you have to insert the appropriate diskette. For example, to start the MS-DOS Shell, you have to insert the SHELL diskette and type DOSSHELL.

You should also create at least one working diskette with the FORMAT command. To do this, type the following command line:

```
FORMAT B:
```

You can then edit files on this working diskette.

From now on you should always use drive B: as your current drive when you work. To do this, insert a working diskette into drive B: and type:

```
B:
```

Now you can proceed in the same manner described in this book. However, you should also include drive A: in the search path so that you don't have to put the drive label in front of all the DOS commands:

```
PATH A:\
```

If you get the error message: "Bad command or filename", you probably don't have the diskette in drive A: which contains the desired command or program. Insert the correct diskette. It's best if you print the contents of the available diskettes and the diskettes created during installation. For more information about that, refer to the Section "Displaying and printing directories".

If the screen prompts you to insert a disk with COMMAND.COM, insert the STARTUP diskette into drive A:.

Appendix C: ASCII Table

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00		32	20		64	40	@	96	60	`
1	01	☺	33	21	!	65	41	A	97	61	a
2	02	●	34	22	"	66	42	B	98	62	b
3	03	♥	35	23	#	67	43	C	99	63	c
4	04	♦	36	24	$	68	44	D	100	64	d
5	05	♣	37	25	%	69	45	E	101	65	e
6	06	♠	38	26	&	70	46	F	102	66	f
7	07	•	39	27	'	71	47	G	103	67	g
8	08	◘	40	28	(	72	48	H	104	68	h
9	09	○	41	29	)	73	49	I	105	69	i
10	0A	■	42	2A	*	74	4A	J	106	6A	j
11	0B	♂	43	2B	+	75	4B	K	107	6B	k
12	0C	♀	44	2C	,	76	4C	L	108	6C	l
13	0D	♪	45	2D	-	77	4D	M	109	6D	m
14	0E	♫	46	2E	.	78	4E	N	110	6E	n
15	0F	☼	47	2F	/	79	4F	O	111	6F	o
16	10	►	48	30	0	80	50	P	112	70	p
17	11	◄	49	31	1	81	51	Q	113	71	q
18	12	↕	50	32	2	82	52	R	114	72	r
19	13	‼	51	33	3	83	53	S	115	73	s
20	14	¶	52	34	4	84	54	T	116	74	t
21	15	§	53	35	5	85	55	U	117	75	u
22	16	▬	54	36	6	86	56	V	118	76	v
23	17	↨	55	37	7	87	57	W	119	77	w
24	18	↑	56	38	8	88	58	X	120	78	x
25	19	↓	57	39	9	89	59	Y	121	79	y
26	1A	→	58	3A	:	90	5A	Z	122	7A	z
27	1B	←	59	3B	;	91	5B	[	123	7B	{
28	1C	∟	60	3C	<	92	5C	\	124	7C	|
29	1D	↔	61	3D	=	93	5D	]	125	7D	}
30	1E	▲	62	3E	>	94	5E	^	126	7E	~
31	1F	▼	63	3F	?	95	5F	_	127	7F	Δ

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	β
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	Ø
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	∈
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	•
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	

Appendix D: Country Codes

DOS can be adapted to the specific requirements of the different countries. For example, you can adapt it to a country's date, time and currency symbol. It's also possible to use country specific characters.

Setting the date and currency

A special driver program is installed through the CONFIG.SYS file to adapt the date, time, decimal separators and currency symbols to different countries. In order to do this, add the following line to your CONFIG.SYS file:

```
COUNTRY=xxx,,COUNTRY.SYS
```

If the COUNTRY.SYS file isn't in the root directory, specify the entire path.

Enter the proper country code, which is displayed in the second column of the following table. For example, you would enter "044" for the United Kingdom. The default settings for DOS install "001", which is the code for the USA.

The country code is taken from the international telephone area codes of the different countries.

Land	Code	Date	Time	Dec.	Tsd.	Code Pg
USA	001	M-D-Y	ST:M:S.HS	.	,	437
French.Can.	002	Y-M-D	ST:M:S,HS	,	,	863
Latin Am.	003	D/M/Y	ST:M:S.HS	.	,	437
Netherl.	031	D-M-Y	ST:M:S,HS	,	.	437
Belgium	032	D/M/Y	ST:M:S,HS	.	,	437
France	033	D/M/Y	ST:M:S,HS	,		437
Spain	034	D/M/Y	ST:M:S,HS	,	.	437
Italy	039	D/M/Y	ST:M:S,HS	,	.	437
Switzerland	041	D.M.Y	ST.M.S.HS	.	,	437
UK	044	D-M-Y	ST:M:S.HS	.	,	437
Denmark	045	D/M/Y	ST.M.S,HS	,	.	865
Sweden	046	Y-M-D	ST.M.S,HS	,	.	437
Norway	047	D/M/Y	ST.M.S,HS	,	.	865
Germany	049	D.M.Y	ST.M.S,HS	,	.	437
Intern.	061	D-M-Y	ST:M:S.HS	.	,	437
Portugal	351	D/M/Y	ST:M:S,HS	,	.	860
Finland	358	D.M.Y	ST.M.S,HS	,		437

Example

For the United Kingdom, the CONFIG.SYS file should contain the following command:

```
COUNTRY=044,,COUNTRY.SYS
```

Country specific special characters

With DOS, the keyboard layout can be adapted to different countries. DOS supports country specific special characters by using code pages (tables of special characters) for the keyboard, screen and printer.

When using different code pages, you must prepare all of your devices according to the manufacturer's specifications. This will ensure that the characters that are printed are the same as those displayed on your screen.

Keyboard

To prepare your keyboard driver to use code pages, type a line similar to the following in your CONFIG.SYS file:

```
COUNTRY=044,Table,COUNTRY.SYS
```

044 is the United Kingdom keyboard layout. If necessary specify COUNTRY.SYS with the complete path. The following parameters can be used for "Table":

437	USA, default
850	multilingual
860	Portugal
863	Canada/France
865	Scandinavian countries

You could use two tables for the United Kingdom (044): 437 and 850.

The following command line is also required for working with code pages. Insert this line in your AUTOEXEC.BAT file:

```
KEYB UK,Table
```

If a value isn't specified for "Table", DOS uses the default table 437.

Depending on the type of installation, DOS could have the following line in your CONFIG.SYS file:

```
INSTALL=C:\DOS\KEYB.COM UK,437
```

This means that DOS was already adapted to the United Kingdom keyboard in the CONFIG.SYS file during installation. So you don't have to enter the line in your AUTOEXEC.BAT file.

Screen

Different code pages can only be used on the screen if the graphics card allows different character displays. Although different code pages can be used with MCGA, EGA and VGA cards, they can't be used with Hercules and CGA cards. Type the following in your CONFIG.SYS file to prepare the screen for code pages:

```
DEVICE=DISPLAY.SYS CON=(Type,Table,Number)
```

If necessary, specify the complete path for DISPLAY.SYS.

Type	EGA for MCGA, EGA and VGA (requires the EGA.CPI file). LCD for LCD screens of portable computers (requires the LCD.CPI file).
Table	Sets the desired code page (see section below).
Number	Specifies the number of desired code pages; later you must choose between them (1 for LCD, 2 for EGA).

Example

Suppose DISPLAY.SYS is in the DOS subdirectory of the hard drive, the computer has an EGA card, and you want to use two code pages with the standard table (USA). Type the following in CONFIG.SYS:

```
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,437,2)
```

You must then make the following change with the MODE command:

```
MODE CON CP PREP=(Table) Type
```

Specify "Table" and "Type" as shown above.

Printer

Code pages can be used on the IBM Graphics Printer II (4201) and the Quietwriter III (5202). They can also be used on compatible printers on a parallel port.

Insert the following line in CONFIG.SYS to use code pages for the printer:

```
DEVICE=PRINTER.SYS LPTx=(Type,Table,Number)
```

If necessary, specify the complete path for PRINTER.SYS. Enter the number of the printer port for x.

Type	4201 (requires the 4201.CPI file). 5202 (requires 5202.CPI).
Table	Sets the desired code page (refer to the section about the keyboard).

Number	Specifies the number of code pages; later you must switch between these pages.

After these steps, you still must use the MODE command:

```
MODE LPTx CP PREP=(Table) Type
```

Specify the "Table" and "Type" as previously shown. For x specify the number of your printer port.

Switching code pages

After adding these lines to your CONFIG.SYS file, you can switch between code pages. However, first you must restart your computer so that these changes can be activated.

Before using code pages, DOS must be modified to support the country-specific character set tables. Use the NLSFUNC (National Language Support Functions) command to do this:

```
NLSFUNC
```

Selecting code pages for the keyboard

You can select another character set table by using the following command:

```
KEYB UK,Table
```

If necessary, specify the complete path for KEYBOARD.SYS.

"UK" represents the United Kingdom keyboard layout. The following parameters can be used for "Table":

437	USA, default
850	multilingual
860	Portugal
863	Canada/France
865	Scandinavian countries

Selecting code pages for the screen

To select another screencharacter set table, you must use two commands:

```
MODE CON CP SELECT=Table
```

"Table" sets the desired code page (see the previous section).

Selecting code pages for the screen in graphics mode

If you're operating your screen in graphics mode, you can display characters with ASCII codes 128 - 255 by using the following command:

```
GRAFTABL Table
```

"Table" sets the desired code page (see the previous section).

Use the following command to determine the status of the graphics characters:

```
GRAFTABL /STATUS
```

Selecting code pages for the printer

Use the following command line to select another printer character set table:

```
MODE LPTx CP SELECT=Table
```

"Table" sets the desired code page (see the previous section).

Avoid switching off a printer when working with code pages (see below).

Setting code pages for all prepared devices

To set a different character set table for all prepared devices, use the CHCP (Change Codepage) command:

```
CHCP Table
```

"Table" sets the desired code page (see the previous section).

If you don't specify any parameters, the command displays the current code page.

Special options for working with code pages

Use the following to obtain information about a device and its defined code page:

```
MODE Device CP /Status
```

"Device" represents the device for which you want information (CON, LPTx). "/Status" can be omitted from the command line.

If device code pages are no longer available, the following reactivates them:

```
MODE Device CP REFRESH
```

"Device" represents the device whose character table set you want to reactivate.

Do this if you switched off the printer while working with code pages.

Appendix E: The Companion Diskette

The companion diskette for this book contains listings for batch, COM and EXE files discussed in this book, to save you the trouble of typing these listings in.

The companion diskette has five subdirectories:

- The BASIC directory contains the QBasic program listings from Chapter 16 (Using QBasic).
- The BAT directory contains all batch file listings presented in this book.
- The DOC directory, which contains an unformatted ASCII file and a PRIVATE directory.
- The SAVER directory contains a Screen Saver program, complete instructions for the screen saver are in the SAVER.TXT file.
- The TIPTRICK directory contains listings from Chapter 17 (MS-DOS Tips And Tricks).
- The TOOLS directory contains a group of new, ready-to-run programs to streamline your sessions with DOS.

Listing of TOOLS programs:

Filename	EXT	Purpose
CLEAR	.BAT	Clears all environment variables
COLOR	.BAT	Configures screen colors
FINDFILE	.BAT	Find files on a storage medium
FINDSTR	.BAT	Searches files for a word
FONT	.BAT	Sets DOS font attributes
GETFUNC	.BAT	Displays function key assignments
HEADMSG	.BAT	Displays parameter as text in inverse video header
HELPME	.BAT	Invokes help screens for companion diskette files
HELPME1	.TXT	Text file read by HELPME.BAT
HELPME2	.TXT	Text file read by HELPME.BAT
HELPME3	.TXT	Text file read by HELPME.BAT
HELPME4	.TXT	Text file read by HELPME.BAT
KEYBOARD	.BAT	Sets keyboard speed
RUNDIR	.BAT	Runs applications from anywhere
SETFUNC	.BAT	Assigns keys to function keys
SFORMAT	.BAT	Formats without danger to hard drive C:
SIZE	.EXE	Determines space free on storage medium
SPRINT	.BAT	Configures printer
TCD	.EXE	Tornsdorf-CD changes to any directory (incl. wildcards)
WAIT	.EXE	Pauses the system or screen display

The TOOLS directory also contains a group of data files, batch files and programs not directly accessible from DOS, but important to the TOOLS files.

Listing of TOOLS utility files:

Filename	EXT	Purpose
CAPITAL	.EXE	Converts characters to uppercase
DIRALL	.EXE	Directory search
DIRDAT	.COM	Directory data
DOSVERS	.COM	Gets DOS version
FINDPRG	.EXE	Program finder
INPUT	.EXE	Input reader
NEXT	.EXE	Next call
PATH_ADD	.DAT	Path addition data
SETINP	.DAT	Input data
SUBPATH	.BAT	Subdirectory path

TINSTALL.BAT

The installation program TINSTALL.BAT will help you install these batch files on your hard drive.

We designed the TINSTALL program to make installation as simple as possible. You can adapt TINSTALL to suit your own needs.

Auto-install from A: to C:\

The companion diskette's \TOOLS directory comprises batch files, help files and other information to make your DOS sessions more efficient. The TINSTALL program was provided to ensure that these files are installed successfully on your system. TINSTALL takes you through the steps needed for installation on drive C:. If you prefer parameters other than (Drive=C:), you have one of two options:

- Change these parameters in the TINSTALL.BAT file using the MS-DOS Editor or another text editor.
- Install the TOOLS data by hand.

Let's look at the steps required.

AUTOEXEC.BAT and CONFIG.SYS

These files can be found in the root directories of almost every PC in existence. AUTOEXEC.BAT and CONFIG.SYS perform some general DOS housekeeping chores (e.g., loading special driver software for mouse or keyboard input). TINSTALL renames these files to AUTOEXEC.OLD and CONFIG.OLD, then generates new versions of AUTOEXEC.BAT and CONFIG.SYS in the system. These new versions have a few lines added that configure your system to the tools on this companion diskette. If you must return to the original files, delete the new

AUTOEXEC.BAT and CONFIG.SYS, and rename the OLD files to their original names.

The backslash "\" and directory specification

When specifying directories you must precede the directory name with a backslash (\), not to be confused with the regular slash (/). Most keyboards have a "\" key just above the Enter key. If your system doesn't have this key, or if your system is configured for another country, you can still enter the backslash using the numeric keypad. Press Alt, then press 9 2.

TINSTALL defaults

TINSTALL defaults to the C: hard drive when installing the tools. TINSTALL also assumes that you are installing the data from drive A:, and that your DOS commands are in a directory named \DOS. TINSTALL also copies the Screen Saver to the TOOLS directory.

For effective installation, you must have COMMAND.COM and ANSI.SYS located in the root directory of your hard drive. If this is not the case, copy these files from your DOS directory. Otherwise, edit the CONFIG.SYS and AUTOEXEC.BAT files after installation to indicate the locations of COMMAND.COM and ANSI.SYS. If you still run into problems, exit TINSTALL and perform the task by hand.

If you wish to change any of these parameters, you must include all four parameters when accessing TINSTALL. These parameters must be entered in the following order:

```
TargetDrv DOSDir SourceDrv
```

Notice that all parameters are separated by spaces.

Here are some examples of calling TINSTALL with parameters:

TINSTALL C: \DOS B:

This runs TINSTALL from drive B:, installing the files to drive C:\, with \DOS as the DOS directory.

TINSTALL C: \DOS5 A:

This runs TINSTALL from drive A:, installing the files to drive C:, with \DOS5 as the DOS directory.

TINSTALL D: \DOS5 A:

This runs TINSTALL from drive A:, installing the files to drive D:, with \DOS5 as the DOS directory.

Specifying drives

Insert the companion diskette in the drive you want to use. Log to A: or B:.

Starting TINSTALL

If the default parameters are suitable (C: \DOS A:), enter TINSTALL and press the Enter key. If you want other information, enter TINSTALL using the options described above and press the Enter key.

Aborting TINSTALL

Once TINSTALL displays the first page of information, you can stop the installation process by pressing Ctrl+Pause (also referred to as the <Break> key). TINSTALL asks if you really want to stop. Press Y to exit.

Once installation is complete you will need to reboot the computer. This puts the information included in the new AUTOEXEC.BAT and CONFIG.SYS files into effect. After installation press Ctrl + Alt + Del to reboot.

Installation by hand

If you choose to install by hand, start by making a directory on your hard drive for the tools using the MD (MKDIR) command. Once you have made this directory, copy the compressed file, TOOLS.COM in the TOOLS directory of the companion diskette to this directory. Then run the TOOLS.COM file and th efiles contained in it will be uincompressed. For example:

```
COPY A:\TOOLS\TOOLS.COM D:\SYSTEM\BATCH
D:\SYSTEM\BATCH\TOOLS
```

This command copies all the files from the TOOLS directory of drive A: to the \SYSTEM\BATCH directory of drive D:.

The path must be changed to give DOS easy access to these files. If you enter the PATH command from within DOS, the current path will appear on the screen. For example:

```
PATH=C:\DOS;C:\APPS\WORD5
```

The path is specified in the file AUTOEXEC.BAT. If you edit the PATH statement in this file then reboot the computer, the system will automatically add this information to DOS, using the INIT.BAT and PATH.TXT files. These two files simply add the path after AUTOEXEC.BAT executes.

To modify the path for access to the sample directories listed above, you would need to delete the INIT.BAT and PATH.TXT files, and add data similar to the following to the end of the PATH statement:

```
;C:\BAT;C:\TIPTRICK;C:\TOOLS
```

before:

```
PATH C:\DOS;\C:\WORD;D:\WORKS
```

after:

```
PATH C:\DOS;\C:\WORD;D:\WORKS;C:\BAT;C:\TIPTRICK;C:\TOOLS
```

While you're editing the PATH statement, add the following data to AUTOEXEC.BAT:

```
@ECHO OFF

SET COMSPEC=C:\COMMAND.COM

SET BatDir=C:\TOOLS

@PROMPT $P$G
```

The first line (@ECHO OFF) suppresses batch file display.

If you install the tools in a directory other than C:\TOOLS, you must set this directory in the SET BatDir= line. Just replace the text C:\TOOLS with the desired directory. Here are two examples in keeping with the sample directories listed earlier:

```
SET BatDir=D:\SYSTEM\BATCH

SET BatDir=C:\TOOLS
```

This information is also needed by the HELPME.BAT file, for access to the WAIT command and the four HELPME*.TXT files.

Insert the following lines in the file CONFIG.SYS:

```
DEVICEHIGH=C:\ANSI.SYS

SHELL=C:\COMMAND.COM /P /E:1000
```

If the DEVICEHIGH=C:\ANSI.SYS or DEVICE=C:\ANSI.SYS commands already exist in CONFIG.SYS, you won't need to repeat that line. The same applies to the SHELL command, but make sure that the existing SHELL command contains the same parameters - our parameters add environment space needed by many of these batch files.

If the ANSI.SYS file is in the DOS directory rather than the root directory, CONFIG.SYS should reflect this in the DEVICE(HIGH) line (e.g., DEVICEHIGH=C:\DOS\ANSI.SYS or DEVICE=C:\DOS\ANSI.SYS). The same applies to COMMAND.COM.

Using these utilities

We'll go into detail about each file soon. For now, here are some ground rules about running the files.

How they work

Most of the files on the companion diskette are batch files - a set of MS-DOS commands, which access MS-DOS commands (e.g., DISKCOPY). These batch files can be discerned by their file extensions of .BAT. The other files are executable programs compiled with Turbo Pascal, distinguished by .EXE and .COM file extensions. These executable files act as support for the batch files - few of them will run on their own, without access through batch files.

There are two methods of running batch files: Direct input from the system prompt, including all batch file parameters; and dialog mode, when provided in the batch file.

What are parameters?

We mentioned batch file parameters earlier. These parameters provide important information to a command or file. Parameters entered in direct access mode must be separated from one another by single spaces. For example, if you wanted to view the directory listing in drive A: of all files, and have this listing displayed in wide (columnar) mode, you would type the following with the /W (wide) switch:

```
DIR A:*.* /W
```

In either the direct access method or dialog mode method, the end result is the same: The batch file acts upon the parameters provided by the user.

Let's take a closer look at both methods, using the COLOR.BAT file, which controls screen colors.

Direct input

Many of the batch files on the companion diskette can be called right from the system prompt without further communication between the user and the system. The syntax of the COLOR.BAT reads:

```
COLOR FOREGROUND BACKGROUND
```

The following changes the screen colors to a red foreground and white background:

```
COLOR RED WHITE
```

Dialog mode

Many of the batch files on the companion diskette also provide a dialog mode, in case users forget a parameter or two. If you enter the COLOR command without any parameters, COLOR displays a dialog box. You can then respond to the prompts displayed by the batch file.

Headers and user information

A batch file named HEADMSG.BAT displays additional information at the top of the screen in a header. Here you may find the name of the file currently running, or text intended to help you execute the batch file. If it does not display properly, make sure your ANSI.SYS file is accessible.

Cancelling and clearing

You can easily exit the dialog mode of a batch file by pressing the Esc key.

However, if you started a batch file from the system prompt in direct mode, you can exit the batch file by pressing either Ctrl+C or Ctrl+Pause (also referred to as the <Break> key). After doing so, run the CLEAR command to ensure that all important system variables are reset to their earlier status. This is especially important if your system prompt has become invisible (it will if @ECHO OFF has been called in a batch file).

Getting general help

Run the HELPME.BAT file. HELPME displays five separate screens listing the batch files available from the companion diskette files. You may prefer to call HELPME from a function key. For more information on assigning data to function keys, see the description of SETFUNC which appears later in this Appendix.

Batch file descriptions

The following paragraphs describe each TOOLS file in detail. The syntax of each batch file is as follows:

COMMAND

Short description

Syntax: COMMAND parameters

Detailed descriptions with examples where applicable.

Here a few rules you should remember about the characters:

- If you plan to enter a path as a File-group parameter, remember to begin the path with a backslash. If you do not do this, the batch file will begin searching at the current directory. Here's an example:

```
\DOCS\PRIVATE
```

- If a path and file identifier is to be entered as a File-group parameter, you must end the directory path with a backslash, immediately followed by the file identifier. The following specifies all DOC files in the \DOCS\PRIVATE directory:

```
\DOCS\PRIVATE\*.DOC
```

- Parameters enclosed in [brackets] are optional, but must be entered if you want to take advantage of this parameter.

- Parameters not enclosed in [brackets] must be entered as part of the batch file. Unbracketed parameters indicate data needed by the batch file.

If you call a batch file using incorrect or missing parameters, the computer beeps and the batch file shifts into dialog mode.

CLEAR

Clears environment variables

Syntax: CLEAR

Pressing Ctrl+C or Ctrl+Pause (also referred to as the <Break> key) during batch file execution stops the batch file, but environment variables may still contain data.

Calling CLEAR resets existing system variables to their original values, or removes variables that may have been added since booting the computer. CLEAR also restores the default prompt if the prompt has been changed since booting the computer.

COLOR

Selecting screen colors

Syntax: COLOR Foreground [Background]

The COLOR program changes foreground and background screen colors. If you enter only the Foreground parameter, the foreground color is affected. If you enter the Background parameter as well, the foreground and background colors will change.

Available colors:

Black
Blue
Red
Magenta (Violet)
Green
Cyan (Light blue)
Yellow
White

FINDFILE

Find file on hard drive

Syntax: FINDFILE Filename

As directory structures become more and more complicated, the problems with keeping track of files increase proportionally. It gets harder to know where a file is on a disk drive, or if duplicates of the same file exist. FINDFILE is the answer.

FINDFILE.BAT searches the current storage medium for a file, informing you of the location of that file in the directory tree.

The Filename parameter specifies the file you want to find. Wildcards are not allowed here - if you do use a wildcard, FINDFILE assumes that all the other characters entered are wild as well.

The following searches for CONFIG.SYS:

```
FINDFILE CONFIG.SYS
```

The following searches for all files using the name/extension/path CON:

```
FINDFILE CON
```

It is also possible to search for files with more than one extension if you enter a period followed by one or more characters. The following searches for every file with an extension beginning with C:

```
FINDFILE .C
```

The following searches for all files with .TXT extensions:

```
FINDFILE .TXT
```

FINDFILE can also be instructed to find filenames immediately to the left of the "dot" and extensions immediately to the right of the "dot". The following searches for files ending with BOOK, and with extensions starting with TX:

```
FINDFILE BOOK.TX
```

A backslash at the beginning of the Filename parameter searches for files beginning with the characters immediately following the backslash. The following searches for files and directories starting with BOOK:

```
FINDFILE \BOOK
```

☞ Do not use wildcards (* or ?) with FINDFILE. FINDFILE will then search for files containing these characters.

The first time you run FINDFILE it will generate a file named FINDFILE.TMP. This contains all the available files. Each time you call FINDFILE, the batch file refers to FINDFILE.TMP on subsequent calls, thus speeding up access time. Entering FINDFILE /INIT starts the TMP file from scratch.

FINDSTR

Searches files for string

Syntax: FINDSTR [Path][File-group] Searchstring

Sometimes it's hard to remember which file contains a certain word. FINDSTR.BAT can help you find that file and word.

The FINDSTR command is followed by a space, the path (when needed), the filegroup and the search string. The Searchstring parameter is case-sensitive (i.e., uppercase and lowercase), but not word-sensitive (i.e., whole words as opposed to parts of words). Here are two examples. If you enter "Man" for the search string (omit the quotes), FINDSTR will find the word "Man" in files, but not "man". On the other hand, FINDSTR finds both "Man" and "LaMancha" (whole word and part of a word).

FINDSTR displays the names of files searched, and text strings if found. When the screen becomes full, FINDSTR displays ---MORE---, indicating that you should press a key to continue.

Wildcards are allowed with FINDSTR. This expands file searching capabilities.

The following searches all DOC files for the word "finance":

```
FINDSTR *.DOC finance
```

You can also enter paths to search in specific directories.

The following searches all the files in the \TEXTS\PRIVATE directory for the word "finance":

```
FINDSTR \TEXTS\PRIVATE\*.* finance
```

☞ It's possible to search an entire diskette or hard drive other than the current drive using FINDSTR, just by adding a drive identifier to the command:

```
FINDSTR A:\TEXTS*.DOC Abacus
```

FONT

Change text display on the screen

Syntax: FONT Parameter

This program changes the text attributes as they appear on the screen.

Available options

N	Normal
K	Blinking
B	Bold
I	Inverse
U	Underline
X	Hidden

☞ The underline attribute changes the foreground color.

These attribute changes apply only to DOS - most applications (e.g., word processors) may use their own special screen drivers, thus overriding attributes set by FONT.

GETFUNC

Display function key assignments

Syntax: GETFUNC

This command displays the function key assignments dictated by SETFUNC. Default DOS assignments (e.g., F3 for retyping commands) will not be displayed.

KEYBOARD

Configures keyboard speed

Syntax: KEYBOARD Speed

Sometimes adjusting the speed of your keyboard is a practical move. Many high-speed PCs actually require slowing keyboard response, because the user's hands cannot keep pace with the PC. This utility lets you adjust the speed of the keyboard repeat on your system.

The Speed parameter has two options:

Fast	Keyboard speed increases to maximum.
Normal	Keyboard speed reverts to normal.

RUNDIR

Start program from any directory

Syntax: RUNDIR Programname [Parameter]

Most applications are placed in their own directories. Before you can start an application, you may have to change to that directory. Frequent use of this application may force you to write a batch file to run the program, or you can use RUNDIR (RUN in own DIRectory).

RUNDIR searches on the current storage medium for an application with a .COM, .EXE or .BAT extension and the specified program name, changes to the directory containing that application and runs the application.

You can also include parameters in RUNDIR which control the way a program runs.

The following runs PCSHELL:

```
RUNDIR PCSHELL
```

The Parameter parameter lets you add switches or filenames to the command call. The following runs Microsoft Word and instructs Word to load the last file edited:

```
RUNDIR WORD /L
```

The following runs Microsoft Word and loads a file named BENZER.DOC from the PRIVATE directory:

```
RUNDIR WORD \PRIVATE\BENZER
```

☞ The PATH command in AUTOEXEC.BAT must be configured to direct RUNDIR to the application.

SAVER

Screen Saver utility

Syntax: SAVER [/A /A- /G /G- /V:xx /Q]

SAVER is a useful little utility program for the MS-DOS 5.0 operating system. It blanks the screen after a pre-determined amount of time. This saves you screen from screen-burn.

SAVER runs on IBM XT/AT or compatible computers with 8088, 8086, 80286, 80386, and 80486 processors. It requires MS-DOS version 3.1 or higher and, under MS-DOS 5.0, works together with the Task Switcher. You get the best results with a VGA screen, but CGA or EGA also work. You have to start the screen driver, MSHERC.COM (with MS-DOS 5.0) to use it a Hercules cards. Since SAVER swaps the contents of the screen and parts of the program memory and has to download program sections, it's best if you install it on the hard drive. You should have at least 500K available memory there. If you have expanded memory in accordance with EMS 3.0 or extended memory in accordance with XMS 2.0, the program automatically makes use of it. Since you only need this memory when the screen is switched off, this doesn't place any limitations on your other programs. You still have as much memory as you did before.

The following switches are supported:

/A	corresponding to the check box, "Saver Active", activates screen blanking in accordance with the set delay time.
/A-	deactivates screen blanking.
/G	corresponds to the "Display Graphics" check box, causing the fireworks graphic to display during screen blanking.
/G-	suppresses this graphic, the screen is dimmed.
/D:xx	sets the delay time. xx is the period in minutes and ranges from 1 to 60 minutes.
/Q	quits SAVER and releases the memory which the program was taking up. As in interactive mode, here too you cannot quit the program at any time. If the start-up message reappears after you call SAVER /Q, it means it wasn't possible to quit the program.

When you start SAVER it is installed as a TSR program. The program only takes up 6K of main memory. At first you won't even notice its presence. SAVER remains entirely in the background. It does monitor the keyboard and mouse, as soon as the computer exceeds a limit specified by the user, SAVER switches off the screen. To prevent others from thinking that the computer is switched off and switching off the computer, fireworks are dispalyed on the screen. As soon as you resume your work on the computer and press the keyboard or mouse, the previous screen contents reappear. continue working. This doesn't just work at the DOS level. SAVER can interrupt any DOS program in order to display the "fireworks". You don't lose any data while this is happening. After reactivating the screen, you can continue working as though nothing happened.

SAVER consists of several files. The TINSTALL.BAT file will install SAVER into the TOOLS directory. For manual installation, simply copy the following files to any directory of your hard drive:

```
SAVER.EXE     SAVER.FON     _BS.0     _BS.1     _BS.2
```

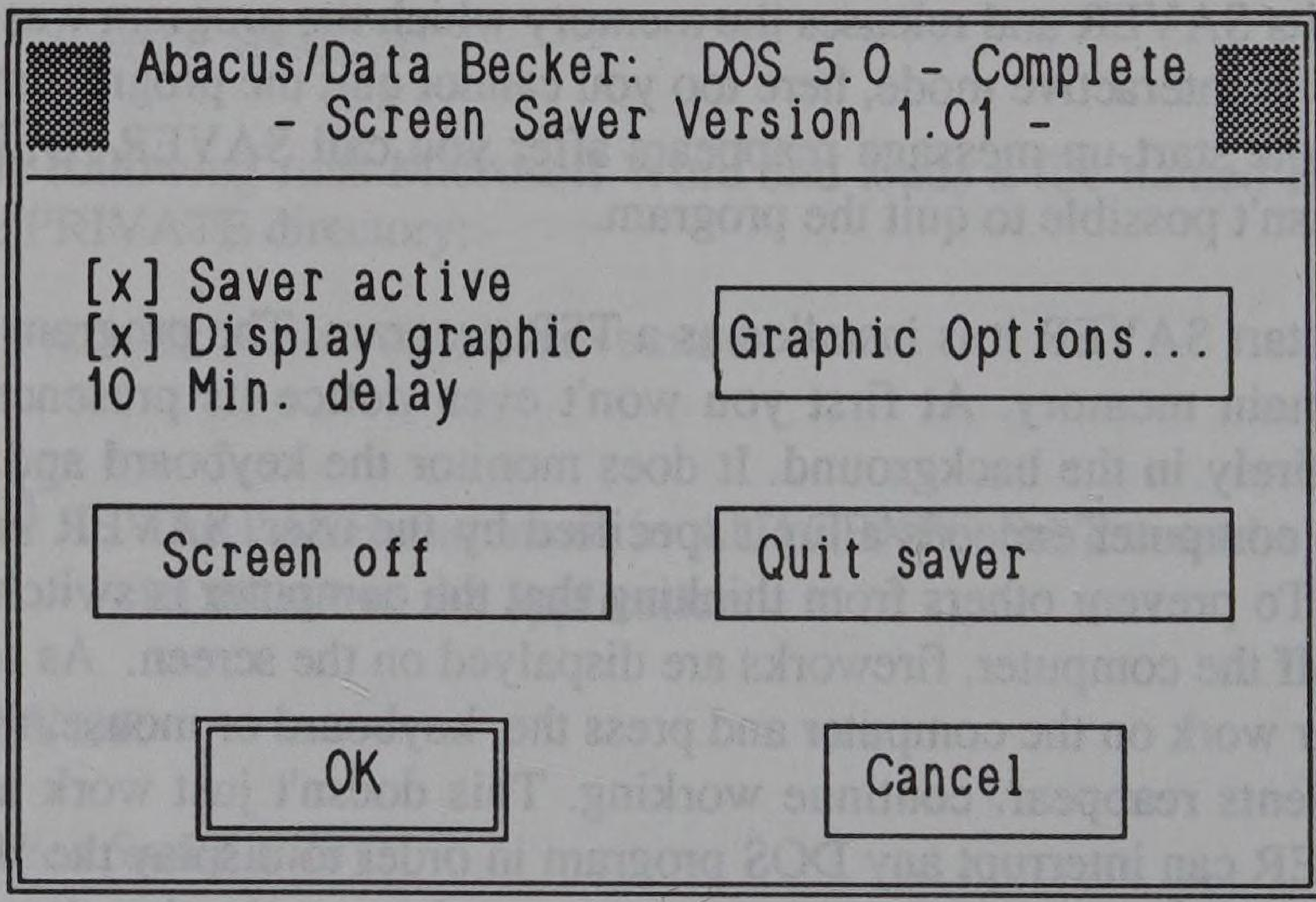
To start the program, simply call "SAVER". You can also include this in the AUTOEXEC.BAT file, so that SAVER is available whenever you reboot the computer. It's enough to enter the following line in the AUTOEXEC.BAT file:

```
[D:\PATH\]SAVER rem >NUL
```

To suppress the start-up message that displays when you start the screen saver, simply remove the "rem" from this line. You need to specify the drive and the path if SAVER is not in the DOS search PATH. The SAVER directory does not have to be in the access path in order for SAVER to find its files. However, if the SAVER directory is not in the access path, you have to specify its path when you call it.

Please make sure that you start SAVER only on the DOS level, and never from an application program or from the DOS-Shell (e.g., "Library" - "Operating system" of MS-WORD). Starting SAVER from a program can cause your system to crash!

You can operate SAVER by typing the command line or from a dialog box. You can call the dialog box any time by pressing the Alt key, the Ctrl key and the left Shift key at the same time. The current program is interrupted and the Screen saver dialog box is displayed.



Abacus/Data Becker: DOS 5.0 - Complete
- Screen Saver Version 1.01 -

[x] Saver active
[x] Display graphic
10 Min. delay

At first, the "Saver active" check box is displayed in inverse video, indicating that it is the currently active box. Press Tab to move through the dialog box (press Shift+Tab to move backwards). Instead of using Tab, you can also select single settings by holding down the Alt key and typing one of the letters that are bolded.

Pressing Alt+G would take you to the check box where you can switch on/off the fireworks display. Pressing the space bar switches the currently active element on or off.

Dialog box options:

Saver Active:	When this box is checked, the screen saver starts working after the set delay, otherwise the screen stays on. This is useful for preventing SAVER from interrupting a program. No other programs can run while the screen is turned off!
Display Graphic:	When this box is checked, the screen dims and then displays the fireworks, otherwise the screen remains black. Here too, the currently running program is interrupted!
Min. Delay:	Here you indicate how many minutes should pass without using the keyboard or mouse before SAVER becomes active. The minimum is 1 minute, the maximum is 60 minutes.
Screen Off:	You can select this option to turn off the screen immediately without having to wait for the set delay time. Depending on the options you have selected, either the fireworks is displayed or else the screen is dimmed.
Graphic Options:	When you press the space bar (or Alt+O) to change the status of this field, a second dialog box appears, where you can make additional settings for the fireworks graphic.
Quit Saver:	This not only closes the dialog box, but also the removes resident part of SAVER. The memory used up by SAVER is released. Please note that it's not always possible to remove a memory resident program. If it's not possible to quit the program, a message will be displayed.
OK:	This box accepts the settings that you made in the dialog box and returns to the interrupted program. SAVER remains installed. As an alternative to pressing the space bar, you can also press Enter to get this function.
Cancel:	This box cancels all of the settings that you made in the control program and returns to the interrupted program. Instead of pressing the space bar, you can also get this function any time by pressing Esc.

You can control SAVER via batch files. For example, before starting a program that you know will run for a long time without any keyboard input, you could deactivate the screen saver to prevent it from interrupting the work of your program. You could create the following batch file to do this for you:

```
@ECHO OFF
[D:\PATH\]SAVER /A- >NUL
program    REM Calls your program
[D:\PATH\]SAVER /A >NUL
```

You only need to specify the drive and the path if the SAVER directory is not in the current path which you set with PATH. The character sequence ">NUL" suppresses SAVER's start-up message by redirecting it to the "NUL device".

Screen Saver Technical notes

SAVER works transparently in the background, i.e., without influencing your work on the computer. Since all of the settings that you make in the control program remain in effect even after you switch off the computer, you usually don't need to worry about the operation of the program. However, it is useful to remember the following two points.

Quitting the screen saver

It's not always possible to quit SAVER and release the occupied memory. This is not just a problem for SAVER, but holds true for all resident programs. You won't be able to quit SAVER after starting other resident programs. You also can't quit the program when you call COMMAND.COM (e.g., LIBRARY-OPERATING SYSTEM of MS-WORD) from another program.

Compatibility with other programs

You should be especially careful with programs that access hardware. Programs that have to do with reorganization of the hard drive fall into this category. You should only run programs for changing the interleave factor, defragmentizers such as Norton's SpeedDisk or similar utility programs after removing all of the resident programs from the memory. At the very least, deactivate SAVER while you work with such programs. Usually the manufacturers of such critical programs bring this to your attention in their manuals.

A number of programs make extensive use of the capacities of the graphics hardware of your computer. Games and some graphics programs, especially those that allow you to change the color palette of the graphics adapter, fall into this category. If SAVER interrupts such a program, the contents of the screen are not are restored correctly when you reactivate the program. In such cases it's enough to deactivate SAVER temporarily before starting the program.

Since SAVER makes use of a simple form of multitasking (running more than one program at the same time on only one processor), problems can arise with other software that also uses multitasking. We don't recommend running SAVER with WINDOWS 3 since it is only possible within limitations:

Basically, you should not run more than one DOS application at the same time. Switching between the applications can lead to a system crash! In such cases, quit SAVER before calling WINDOWS.

In "Enhanced 386" mode of WINDOWS 3 you can run DOS applications in the background while working with another program in the foreground. If you want to make use of this option, deactivate SAVER before calling WINDOWS. This way you prevent SAVER from trying to turn off the screen while you are working with a WINDOWS application. Programs running in the background get neither keyboard nor mouse input. SAVER would believe that you were no longer working on the computer in such a case.

As long as WINDOWS 3 is running, SAVER is only active when the computer is in text mode. The screen saver goes into action during DOS applications only when they are not running in graphics mode. SAVER does not interrupt WINDOWS applications.

The easiest way to temporarily deactivate SAVER is to create a batch file for starting WINDOWS with the following content:

```
@ECHO OFF
[D:\PATH\]SAVER /A- >NUL
win
[D:\PATH\]SAVER /A >NUL
```

If you are planning to use several DOS programs at the same time, quit SAVER entirely before starting WINDOWS:

```
@ECHO OFF
[D:\PATH\]SAVER /B
win
```

Of course, you only have to specify the drive and path if you didn't use PATH to include the SAVER directory in the access path.

SETFUNC

Function key setup made easy (F1-F10)

Syntax: SETFUNC Function-key [Assignment] [/RETURN]

The MS-DOS function key allocations don't allow for much flexibility. SETFUNC (SET FUNCtion keys) lets you give new assignments to function keys, whether these assignments access DOS commands or call applications.

Entering SETFUNC Function-key without any other parameters clears that function key of its current setting.

The following clears any settings from F3:

```
SETFUNC F3
```

SETFUNC with an Assignment parameter designates the specified function key as having that assignment. Pressing the function key and the Enter key executes that command.

The following assigns the DIR command to the F3 key:

```
SETFUNC F3 DIR
```

The /RETURN switch adds the ability to execute commands immediately, as if you'd entered the command by hand and pressed the Enter key.

The following assigns the DIR command to the F3 key and adds the Enter key for immediate command execution:

```
SETFUNC F3 DIR /RETURN
```

SETFUNC concludes with a listing of current function key assignments.

☞ These function key assignments are lost when you reboot the computer. If you would like to have permanent function key re-assignments, you can add the definitions in your AUTOEXEC.BAT file (see your DOS documentation for more information on MS-DOS, or read *MS-DOS for Beginners*, published by Abacus). AUTOEXEC.BAT requires a CALL command to access SETFUNC. Inserting the following line in AUTOEXEC.BAT calls SETFUNC and makes the F4 key the "official" DIR key:

```
CALL SETFUNC F4 DIR /RETURN
```

SIZE

Returns memory used by selected files

Syntax: SIZE [Path][File-group]

It is often necessary to know how much space a selected group of files takes on a disk. This can be useful when you plan to copy the contents of a subdirectory to a diskette, but you want to be sure that enough space exists on the diskette to complete the copy operation. The SIZE command is useful for this.

If you enter SIZE without a File-group parameter, SIZE checks the file allocation for the entire storage medium on the current drive and directory. It displays how many bytes would be needed for transfer to a 360K (or 720K) diskette, and for transfer to a 1.2 Meg (or 1.44 Meg) diskette.

The differences between normal and high-density diskettes appear because of the way MS-DOS reserves memory on varying diskette types.

SIZE provides the option of entering a path and File-group to limit the file check.

The following checks the amount of memory occupied by the DOC files in the TEXTS directory of the current drive:

```
SIZE \TEXTS\*.DOC
```

SFORMAT

Format floppy diskettes

Syntax: SFORMAT Drive: [Options]

The FORMAT command is an important but dangerous command. As it will format any storage medium, there is the possible danger of accidentally formatting a hard drive or diskette, thus destroying valuable data.

The SFORMAT (Safe FORMAT) command will not allow access to drive C: for formatting. If you enter C: as the Drive: parameter, the computer beeps and SFORMAT informs you that you cannot format the hard drive from within SFORMAT.

The Drive: parameter specifies the drive you want used for the format. Type the drive identifier followed by a colon.

The following invokes SFORMAT for formatting drive A:

```
SFORMAT A:
```

The Options designate the type of formatting you want performed. The following switches are available:

/Q	Performs a quick format (a new feature added to MS-DOS 5.0).
/V	Allows the entry of a volume label. The /V switch must be immediately followed by a colon and the volume label name (e.g., /V:RALPH).
/U	Unconditional format (no UNFORMAT possible).
/4	Formats a diskette as 360K (XT format). Available only to AT users.
/B	Allocates space on the storage medium for system files.
/S	Copies system files to the diskette, making it a bootable (system) diskette.

The following formats the diskette in drive A:, assigns a volume label of BOOTDISK and makes the disk a system diskette:

```
SFORMAT A: /V:BOOTDISK /S
```

SPRINT

Configure printer parameters

Syntax: SPRINT Parameter

This program configures a few printer parameters (e.g., character height or line height) for printing under MS-DOS.

Remember the following when using this program:

- If you call SPRINT in direct mode, you have the option of entering a parameter for line height as well as character height.

- In dialog mode you can only enter one parameter per execution of the program. The following parameters are available in SPRINT:

Character size

Wide:	Printing at 80 characters per line
Compressed:	Printing at 132 characters per line

Line height

Tall:	Printing at 6 lines per inch
Short:	Printing at 8 lines per inch

Default parameter

Normal:	80 characters per line, 6 lines per inch

This program will only work under MS-DOS (e.g., printing directories). Applications such as word processors usually contain their own special printer configuration commands.

TCD

Fast directory change with wildcards

Syntax: TCD Directoryname

Complicated directory structures are important for maintaining order on your hard drive. It becomes tiresome after a while, however, to enter complete pathnames every time you wish to change directories with the CD command. The TCD (Tornsdorf CD) command offers an alternative.

TCD lets you switch to any directory on the current disk without your having to enter the full pathname. It works similarly to the Norton Utilities NCD command, except that TCD doesn't require a companion file. This saves you the trouble of updating the companion file every time the directory structure changes.

You can even use wildcards with TCD. For example, TCD DB* will change to the DBASE directory from any other directory on the storage medium.

TCD searches the hard drive for a directory with the specified name and changes to it. This works even if the directory name you enter is a subdirectory of another directory.

Use TCD as you would the CD command. Unlike the CD command, however, TCD can only be used on the current drive. You may want to place a copy of TCD on each hard drive available on your system.

The following changes to the next highest directory in the directory tree:

```
TCD
```

The following changes to the root directory:

```
TCD \
```

The following changes to a directory name starting with PRIV (e.g., C:\DOCS\LETTERS\PRIVATE):

```
TCD PRIV*
```

☞ Avoid using duplicate directory names in different subdirectories of a disk. TCD will find only the first available occurrence, ignoring any subsequent occurrences.

Appendix F: QBasic Keywords

QBasic is a rich and powerful language, as is demonstrated by the following list of keywords. To view all QBasic keywords from the QBasic Editor, select **Help**, then choose the **Index** command. To obtain information about a specific keyword, place the cursor in the word, then press F1.

An alternate method of obtaining information is to type the keyword in the Editor, move the cursor into the keyword and press F1. The Help screen for the keyword will be displayed.

$DYNAMIC
$STATIC
ABS
ABSOLUTE
ACCESS
AND
APPEND
AS
ASC
ATN
BASE
BEEP
BINARY
BLOAD
BSAVE
CALL
CALL ABSOLUTE
CASE
CDBL
CHAIN
CHDIR
CHR$
CINT
CIRCLE
CLEAR
CLNG
CLOSE
CLS
COLOR
COM
COMMON
CONST
COS
CSNG
CSRLIN
CVD
CVDMBF
CVI
CVL
CVS
CVSMBF
DATA
DATE$ Function
DATE$ Statement
DECLARE
DEFDBL
DEFFN
DEFINT
DEFLNG
DEFSEG
DEFSNG
DEFSTR
DIM
DO...LOOP
DOUBLE
DRAW
ELSE
ELSEIF
END
ENVIRON
ENVIRON$
EOF
EQV
ERASE
ERDEV
ERDEV$
ERL
ERR
ERROR
EXIT
EXP
FIELD
FILEATTR
FILES
FIX
FOR...NEXT
FRE
FREEFILE
FUNCTION
GET(FileI/O)
GET(Graphics)
GOSUB
GOTO
HEX$
IF..THEN...ELSE
IMP
INKEY$
INP
INPUT
INPUT$
INSTR
INT
INTEGER
IOCTL
IOCTL$
IS
KEY(Assignment)
KEY(EventTrapping)
KILL
LBOUND
LCASE$
LEFT$
LEN
LET
LINE(Graphics)

LINEINPUT
LIST
LOC
LOCATE
LOCK...UNLOCK
LOF
LOG
LONG
LOOP
LPOS
LPRINT
LPRINT USING
LSET
LTRIM$
MID$ Function
MID$ Statement
MKD$
MKDIR
MKDMBF$
MKI$
MKL$
MKS$
MKSMBF$
MOD
NAME
NEXT
NOT
OCT$
OFF
ON
ON...GOSUB
ON...GOTO
ON COM
ON ERROR
ON KEY
ON PEN
ON PLAY
ON STRIG
ON TIMER
OPEN
OPENCOM
OPTION BASE
OR
OUT
OUTPUT
PAINT
PALETTE
PCOPY
PEEK
PEN
PEN
PLAY
PLAY(Event Trapping)
PLAY(Music)
PMAP
POINT
POKE
POS
PRESET
PRINT
PRINT USING
PSET
PUT(FileI/O)
PUT(Graphics)
RANDOM
RANDOMIZE
READ
REDIM
REM
RESET
RESTORE
RESUME
RETURN
RIGHT$
RMDIR
RND
RSET
RTRIM$
RUN
SCREEN Function
SCREEN Statement
SEEK Function
SEEK Statement
SELECTCASE
SGN
SHARED
SHELL
SIN
SINGLE
SLEEP
SOUND
SPACE$
SPC
SQR
STATIC
STEP
STICK
STOP
STR$
STRIG Function
STRIG Statement
STRING
STRING$
SUB
SWAP
SYSTEM
TAB
TAN
THEN
TIME$ Function
TIME$ Statement
TIMER Function
TIMER Statements
TO
TROFF
TRON
TYPE
UBOUND
UCASE$
UNLOCK
UNTIL
USING
VAL
VARPTR
VARPTR$
VARSEG
VIEW
VIEWPRINT
WAIT
WEND
WHILE...WEND
WIDTH
WINDOW
WRITE
XOR

Appendix G: Quick Index

MS-DOS 5.0 Commands

APPEND 659
ASSIGN 662
ATTRIB 663
BACKUP 665
BREAK 667
CALL 668
CD 669
CHCP 670
CHDIR 671
CHKDSK 672
CLS 673
COMMAND 674
COMP 676
COPY 677-679
CTTY 680
DATE 681
DEBUG 682
DEL 690
DELOLDOS 691
DIR 692
DISKCOMP 695
DISKCOPY 696
DOSKEY 698
DOSSHELL 701
ECHO 702
EDIT 703
EDLIN 704
EMM386 706
ERASE 707
EXE2BIN 708
EXIT 709
EXPAND 710
FASTOPEN 711
FC 712
FDISK 713
FIND 715
FOR 716
FORMAT 717
GOTO 720
GRAFTABL 721
GRAPHICS 722
HELP 724

IF 725
JOIN 727
KEYB 728
LABEL 731
LH 732
LOADFIX 733
LOADHIGH 734
MD 735
MEM 736
MIRROR 741
MKDIR 742
MODE 743-747
MORE 749
NLSFUNC 750
PATH 751
PAUSE 752
PRINT 753
PROMPT 755
QBASIC 757
RD 758
RECOVER 759
REM 760
RENAME 761
REPLACE 762
RESTORE 764
RMDIR 766
SET 767
SETVER 769
SHARE 771
SHIFT 772
SORT 773
SUBST 775
SYS 776
TIME 777
TREE 778
TYPE 779
UNDELETE 780
UNFORMAT 781
VER 782
VERIFY 783
VOL 784
XCOPY 785

Index

3 1/2 inch disk drive ... 88, 96-97
  See also Disk drives
  1.44 Meg ... 89, 169
  2.88 Meg ... 89
  720K capacity ... 89, 169
  Capacity ... 96
  Illustrated ... 89
5 1/4 inch disk drive ... 95-96
  See also Disk drives
  1.2 Meg ... 168-169
  360K disk drive ... 168
  Protecting ... 87
  Write protect notch ... 87

A

Abort, Retry, Fail ... 606
Access denied ... 239
Address lines ... 384
Allocation unit ... 142
Alphanumeric variables ... 584
ANSI.SYS ... 366, 369, 371, 601
  Example ... 788-791
  Programming keyboard ... 370
  Screen control ... 366
APPEND ... 190, 291, 348, 354
Application programs:
  See also programs
  Avoiding problems ... 309-313
  Exiting ... 20-21
  Exiting in MS-DOS Shell ... 37-38
  From diskette ... 91
  From hard drive ... 91-92, 290-292
  From MS-DOS Shell ... 293-304
  Installing ... 282-289
  Quitting ... 308
  Starting ... 290
  Using MS-DOS ... 279-313
  With parameters ... 305-306
Archive flag ... 223, 239, 240
ASCII Table ... 603, 808-809
ASSIGN ... 146, 147, 148
ATTRIB ... 239, 243, 631, 645
  Changing archive flag ... 239-240
  Changing hidden attribute ... 241-242
  Changing system attribute ... 242-243
  Changing write protect flag ... 239
  Displaying file attributes ... 243
  Marking files ... 240-241
Attribute ... 239
AUTOEXEC.BAT ... 215, 345, 602
  Load MS-DOS Shell automatically ... 56
  Setting up ... 52-56
  Tips and hints ... 56

B

Background printing ... 233
Backslash ... 177, 183
BACKUP ... 215, 239-240, 253
Backups:
  Automatic ... 265-267
  BACKUP Command ... 215, 239
  Backup copy ... 253
  BACKUP.LOG ... 260
  By DATE ... 260
  Hard drive ... 253-254
  By date ... 261
  Changed files ... 258, 259
  Diskettes required ... 625
  Frequency ... 254
  Options ... 260-261
  Reasons to backup data ... 251- 252
  Selected files ... 257-262
  Specific files ... 257-258
Batch files ... 449-485
  .BAT extension ... 450
  Aborting ... 457
  Calling a batch file ... 473-474
  Changing ... 459, 460
  Character strings ... 464
  Characters to use ... 482
  Commands ... 461-476
  Creating ... 453-455
  Displaying ... 458
  DOSKEY ... 483-484
  Editing with EDLIN ... 405-412
  Environment variables ... 480-481
  From MS-DOS Shell ... 517
  Interrupting ... 475
  Looping ... 472-473
  Parameters ... 477-479
  Passing parameters ... 476
  Printing ... 458-459
  REM statement ... 463
  Running ... 456
  Similarity to programs ... 450-452
  Special features ... 369
  Starting ... 456
  Testing conditions ... 464
  Using % ... 482
  Using < ... 482

- Using >........482
- Using batch files........456-457
- Using comments........463
- Using ECHO........461-463
- Using EXIST........465
- Using |........482

Baud rate........362

Beep........634

Boot diskette........106

Boot sector........166, 167

BREAK........54, 355

Buffer size........233

BUFFERS........357

C

Cache........389

Cache program........154, 158-159, 389

CALL........473

CD........11, 182-183, 611

CGA........364

Chains........143

Change Directory........11

Changed file........258

Changing file attributes........547

Character sets........52

Character strings........584

CHKDSK........215, 245-247, 273, 621
- Correcting errors........144-145
- Displaying memory........142
- Information on errors........143
- Locating the error........144
- Recognizing errors........141

CLEAR........822

Clearing the screen........316

CLS........316

Cluster........275

Colon........471

COLOR........822

Color graphics adapters........364
- CGA........364
- EGA........364
- VGA........364

Command interpreter........57, 290-292
- Second command interpreter........349-351
- Specifying........345-348
- Using the command interpreter........349-351
- Using variables........347-348

Command line........2-5, 24
- Displaying........356
- Entering commands........67
- Jumping........471-472
- Using........5

Command word........561

COMMAND........349-351

COMMAND.COM........57, 308, 531
- Invalid........347
- Specifying command interpreter........345
- Tips on using........601, 610

COMP........225

Companion Diskette........815

Computers:
- Error message........48-50
- Language........561
- Problems........50

COMSPEC........53, 308, 354, 610

CONFIG.SYS........52, 215, 366, 401, 601
- Currency format........56
- Date format........56-57
- File........56
- Sample........61
- Setting up........56-61
- Using SHELL command........345-346

Contiguous data........621

Contiguous files........245

Conventional memory........58-59, 384-386

COPY........12-15, 218, 458, 620
- From diskette to hard drive........14-15
- From hard drive to diskette........13-14

COPY CON........602
- Creating text files........215
- Editing files........413
- Special options........221

Copying
- *See also COPY and DISKCOPY*
- Checking copies........220
- Copying from a directory........218-219
- Copying more than one file........219-220
- In MS-DOS Shell........28-32
- Installation programs........282
- Single disk drive........137-138
- Two disk drives........138
- Using XCOPY........222-224
- Verifying........138, 224

COUNTRY........56

Country codes........810-814

Current Directory........80, 182

Current input device........315

D

Data bits........362

Data verification........355

Data, Contiguous........621

Data, Sorting........320

DATE........5, 260, 309, 336-338

Date:
- Creation date........336

  Format....................56-57
  Using DATE..........5, 260, 309, 336-338
DEBUG........................636
DEL..................21, 236, 645
DEVICE........................58
Device name......................176
Device drivers....................787-801
DEVICEHIGH..................55, 61, 396
DIR:
  Default switches....................200
  Display any directory....................196
  Display by attributes....................194
  Display directory..............132, 193-200
  Display diskette directory..............9-10
  Display five columns....................197
  Displaying files....................7-11
  Displaying page by page....................9
  Displaying subdirectories............8, 199
  Special features....................199
  Using in printouts....................200
  Using wildcards..............133,198-199
DIRCMD....................348, 354, 617
Directories....................27, 173-174
  See also Directory
  Changing directories..............611, 644
  Defined....................46-47
  Deleting....................207
  Displaying structure....................201
  Removing....................207-208
  Renaming....................205-206
Directory....................149, 193-195
  See also Directories or Root directory
  Change batch files....................612
  Changing..............10-12, 182-183
  Choosing....................182
  Copying several....................222-224
  Current directory..............80, 182-184
  Data files....................285
  Date and time....................9
  Defined....................46, 174
  Directory names..............176, 181
  Displaying..........9, 131-135, 184, 195
  Empty....................208, 222
  Management....................196
  MS-DOS Shell....................547
  Number of files....................9
  Program files....................285
  Protecting....................618
  Pseudo-drive....................209
  Removing....................207-208
  Renaming....................205-206
  Sequence....................651
  Serial number....................8
  Sorting....................333
  Specification....................8
  Storage space....................9
  Subdirectory....................8
  Using filers....................19
DISKCOMP....................139-140
  Single disk drive....................139-140
  Two disk drives....................140
DISKCOPY.............. 23-24, 620, 652
  Single disk drive....................137-138
  Two disk drives....................138
  Verifying copy....................138
Disk Directory....................7, 162
Disk drives:
  See also specific types
  Additional drives....................61
  Capacities....................168
  Defined....................48
  Defining capacity....................61
  Defining disk drive....................379-380
  Designator....................7
  Determining space....................625
  Different types....................168
  Installing new disk drive..........378-379
  Selecting....................131
Diskette:
  3 1/2" diskettes....................88-89
  5 1/4" diskette....................87-88
  Boot block....................166-167
  Comparing....................139
  Copying..............23-24, 136-138
  Defined....................48
  Different capacities....................102-104
  FAT....................163-165
  Formatting..............22-23, 101
  Number for backup....................625
  Sectors....................161
  System diskettes....................169
  Table of contents directory..............162
  Tracks....................161
  Verify copying....................138
  Write protect notch....................24
Display mode:
  Setting....................364
  Text lines....................365
DISPLAY.SYS....................792
DOS Commands *(see MS-DOS Commands)*
DOS=HIGH....................400
DOS=LOW....................400
DOSSHELL....................491
DOSKEY....................69-78
  Buffer....................70
  Creating batch files..............72-73
  Creating macros....................73-77
  Editing batch files..............72-73

  Editing command line 70-72
  Editing keys 77-78
  Exiting 78
  History buffer 70
  Insert mode 69
  Keyboard shortcuts..........................66
  Repeating command line..................71
  Overstrike mode 69
  Running...69
  Using macros.............................73-77
  Using with batch files.....................483
DRIVER.SYS 793

E

ECHO 461, 602
  @ character 461
  ECHO OFF 356
  ECHO ON....................................356
  Using in batch files.........................462
EDIT...425
  See also MS-DOS Editor
  Creating batch files.........................455
EDLIN.......................................405-419
  As text editor...........................413-414
  C (Copy) command...................416-417
  Changing batch files.......................459
  Command prompt 409
  Creating batch files...................453-454
  D (Delete) command.....408, 412, 418, 460
  E (End Edit) command....................412
  E (Exit) command...........................460
  Editing lines..............................406-408
  File maintenance 413
  I (Insert) command 408, 412
  Information..............215, 371-372, 602
  Inserting lines 411, 418
  L (List) command..............406-412, 459
  Line editor 405
  M (Move) command.........................416
  Printing 414
  R (Replace) command...............415-416
EGA graphics adapter 364
EGA.SYS 794
EMM386 Memory manager 55
EMM386.EXE......................392, 400-401
EMM386.SYS............................795-796
Empty directory..........................208, 222
EMS 390
End of file mark...................................221
Environment memory 351-353
Environment space...............................610
Environment variable:
  APPEND 354
  Changing 352-353
  COMPSPEC 354
  Defining 352, 353
  DIRCMD...354
  Displaying 352
  PATH 353
  PROMPT...354
  Special variables..............................353
  TEMP..354
  Using in batch files...................480-481
EOF (End of File)..................................221
ERRORLEVEL 465-471
  Defined..466
  Determining...............................466-467
  Information provided........................467
  Purpose...466
  Syntax...466
  With BACKUP...........................467-468
  With DOS commands.................468-471
Errors 465, 636
  Bypassing error messages..................606
  On storage devices............................143
Escape sequence 366
EXIT 345, 504
Expanded Memory 58-59, 390-392
  Cache programs 158
  EMS...390
  Layout..391
  Utility programs 392
Extended DOS partition.........................112
Extended memory 58-61, 384-389
  As a RAM disk...........................155-156
  Cache programs 158-159
  Disadvantages 390
  EMS...399
Extension................................82-85, 541

F

FASTOPEN 58, 159, 249
FAT 163-165, 275-276
FC 225-230
FDISK 113
File Allocation Table *(see FAT)*
File extensions:
  See also Files
  .BAK..212
  .BAS 213
  .BAT 212
  .COM...212
  .DBF 212
  .DIC 212
  .EXE 212
  .PAS 213

    .SYS 212
    .TMP 213
    .TXT 212
File:
    See also Files
    Archive flag 239
    Attributes 239
    Attributes, Displaying 243
    Defined 45-46
    Displaying by pages 317
    Extension 45, 82-85, 541
    File date 336
    File ending 213
    Renaming in MS-DOS Shell 547
Filename 580
    Allowed 84, 211
    Defined 45
    In MS-DOS Shell 26
    Using in DOS 82-84
FILES 57, 358
Files:
    Archive flag 240-241
    Backup by date 260-261
    Backup changed files 258-259
    Backup reasons 251-252
    Backup selected files 257
    Backup specific files 257-258
    Changed 263
    Checking 220, 244
    Combining files 222
    Comparing 225-230
    Contents 248
    Contiguous files 245
    Copying 218-221
    Copying file groups 629
    Copying in MS-DOS Shell 28-32
    Copying several 222-224
    Created by MS-DOS 215
    Creating 215
    Creating by redirection 216-217
    Defined 45-46
    Deleting (in MS-DOS Shell) 38-39
    Deleting 21, 236-237
    Deleting all 236
    Differences 226
    Displaying 231
    Displaying certain files 198
    Displaying in MS-DOS Shell 26-27
    Excluding from commands 630
    File extensions 211-213
    Finding 190
    Moving files 544
    MS-DOS Shell 534-535, 548
    Naming 211-214
    Number opened 358
    Opening faster 249-250
    Printing 233-235
    Protecting 239, 269
    Recognizing files 45-46
    Recovering 255-256, 272
    Renaming 238
    Restoring 255-256, 263-264
    Restoring deleted files 275
    Searching for a file 247
    Searching for file contents 248
    Searching for files 605
    Sorting 320, 551, 638
    Undeleting 237, 276-277
    Using COPY CON 215
    Using EDLIN 216
    Using MS-DOS Editor 216
    Using RECOVER 273-274
    Using wildcards 213-214
    Viewing 231-232
Filters 19, 315, 320, 333
FIND 323, 327, 248, 607
FINDFILE 823
FINDSTR 823
FONT 824
FORMAT 22, 101, 168
Formatting:
    1.2 Meg disk drive 102-104
    1.44 Meg disk drive 105-106
    360K disk drive 102
    720K disk drive 104-105
    Automatic formatting 110
    Cancelling disk formatting 619
    Diskette 168
    Hard drive 111, 125-128
    Special switches 108-109
    System diskettes 106
Fragment 246
Frames 602
Function keys 616, 640-644

G-H

GETFUNC 825
GRAFTABL 54
GRAPHICS 54
Graphics card 364
GW-BASIC 562
Hard drive cache 389
Hard drive:
    Acceleration 158-160
    Access time 99
    Active partition 119-120
    Advantages 98

- Appearance ... 97
- Backup ... 253-254
- Cache programs ... 154
- Capacity ... 89
- Conserving Space ... 654
- Defined ... 48
- Deleting partitions ... 121-122
- Deleting logical drives ... 122
- Determining space ... 625
- Differences from diskettes ... 90
- Disadvantages ... 98
- Extended partition ... 116-117
- File Fragmentation ... 245
- Formatting ... 125-128
- Logical drives ... 117-119
- Partitioning ... 111-115
- Partitions ... 654
- Primary DOS partition ... 115
- Programs ... 290-292
- Protecting ... 90
- Special features ... 90
- Speed ... 99
- Storage capacity ... 98

Hardware, Adapting MS-DOS ... 359
Hidden file ... 208
Hidden space ... 618
High memory ... 59, 393-394
HIMEM.SYS ... 59, 393, 400, 797-798

I-L

Input ... 327

Input device ... 315
- Default ... 327
- Defining ... 82

INSTALL ... 282
Installation ... 215
Installation programs ... 281-288
KEYB ... 52-53
KEYBOARD ... 825

Keyboard:
- Change ... 58
- Disabling <Num Lock> ... 622
- Foreign ... 375
- Increasing speed ... 627
- Key codes ... 373-374
- Layout ... 65
- Modifying ... 375
- Programming ... 370
- Setting layout ... 366
- Shortcuts ... 640
- Special keys ... 371
- Using in DOS ... 65-66

LABEL ... 129
Label ... 471
LASTDRIVE ... 57, 146-149, 209
Link commands ... 333
Link several commands ... 333
LOADFIX ... 402
LOADHIGH ... 55, 392, 396
Log file ... 215, 260
Logical drives, Defining ... 118
Loops ... 472
Lost allocation units ... 143, 145
LPT1 ... 331

M-N

Main memory ... 142
MD ... 180
MEM ... 58, 382-383, 401
Memory sharing, problems ... 388

Memory ... 58-59, 290
- *See also specific types of memory*
- Administration ... 399
- CONFIG.SYS ... 58-61
- Conventional memory ... 384-386
- Conventional-increasing ... 55
- Customizing ... 400-402
- Determining amount of ... 382
- Expanded memory ... 58, 390-392
- Extended Memory ... 58, 388-389
- General information ... 381
- High memory ... 59, 393-394
- HIMEM.SYS ... 59
- Low memory ... 385
- Management ... 59, 381
- Partitioning ... 403
- Total memory ... 59
- Upper memory ... 386, 395-398
- User memory ... 385
- Using MEM ... 382-382
- XMS ... 399

MIRROR ... 268-269
MODE ... 317, 360-364, 376, 627
Monochrome display ... 364
MORE filter ... 315, 317, 327

MS-DOS:
- Adapting to your system ... 51
- Application Programs ... 279-313
- History ... 802-807
- Internals ... 161-171
- Quitting ... 64
- Setting up ... 51
- Starting from diskette ... 62-63
- Starting from hard drive ... 63-64
- Switching versions ... 653
- Tips and tricks ... 601-654

- Using keys.........65
- Versions.........653

MS-DOS Commands.........655-785
- Executing from any directory.........79-80
- How commands are processed.........80
- Defining input/output devices.........82
- Avoiding problems.........312
- Problems.........311

MS-DOS Editor.........421-447
- Blocks of text.........436
- Cancelling selections.........438
- Changing a filename.........431
- Changing batch files.........460
- Copying text.........437
- Creating a file.........428
- Cursor keys.........442
- Cursor movements.........441
- Deleting text.........432-437
- Edit menu.........444
- Editing keys.........439-440
- Editing text.........432-434
- Entering text.........432
- File menu.........443
- From MS-DOS Shell.........532
- General information.........421
- Help menu.........446-447
- Help screen.........425
- Help file.........422
- Insert mode.........432
- Inserting files.........437
- Loading files.........426
- Menu bar.........423
- Menus.........424, 442-447
- Mouse.........442
- Moving text.........437
- Overtype mode.........432
- Pasting text.........437
- Pull-down menus.........423
- Quitting.........431
- Saving a block.........437
- Saving a file.........430-431
- Scroll bar.........423
- Search menu.........444-445
- Starting.........425-428
- Starting a new file.........431
- Tabs.........434

MS-DOS Editor Edit menu.........444
- Clear.........444
- Copy.........444
- Cut.........444
- Paste.........444

MS-DOS Editor File menu.........443
- Exit.........443
- New.........443
- Open.........443
- Print.........443
- Save.........443
- Save As.........443

MS-DOS Editor Help menu.........446-447
- About.........447
- Getting Started.........446
- Keyboard.........447

MS-DOS Editor Options menu.........445
- Display.........445
- Help Path.........445

MS-DOS Editor Search menu.........444-445
- Change.........445
- Find.........444
- Repeat Last Find.........444

MS-DOS QBasic *(See QBasic)*

MS-DOS Shell:
- ADD mode.........500
- Advantages.........489-490
- Application programs.........293-304
- Application Shortcut Key.........519
- Attributes.........547
- Basics.........489
- Changing Areas.........496
- Changing directories.........27, 28
- Commands.........517
- Copy diskettes.........532
- Copying a program item.........537
- Copying diskettes.........41-42
- Copying files.........28-32
- Creating a directory.........547
- Deleting files.........38-39
- Directory Tree area.........534
- Directory Tree/file list area.........534
- Disk Copy.........532
- Disk Utilities.........506, 532
- Displaying directories.........28
- Displaying files.........25-26, 534
- Displaying pathnames.........32
- Drive icons.........495
- Exit temporarily.........531
- Exiting.........37, 531
- File list area.........534
- Formatting diskettes.........39-41
- Graphics mode.........492, 522
- Group, changing.........527
- Group, creating.........527
- Group defined.........505
- Group icon.........508
- Group title.........529
- Groups selecting.........508
- Help menu.........559
- Help Text.........529
- Help Text, Defining.........521

Installing 489
Introduction 495-502
Keyboard layout 503-504
Logical drives..............................495
Main group 505
Menu bar 493
MS-DOS Editor, Running..................532
MS-DOS QBasic, running.................532
New program tips.....................523-524
Password 519, 529-533
Program group 528-530
Program group, defined...................508
Program icons...............................507
Program item adding.......................516
Program item changing 517
Program list area......................530-531
Program list, customizing................516
Programs.................................508-509
Quitting 308
Reserve Shortcut Keys.....................523
Return from DOS prompt..................531
Running programs 35-36, 490
SAA Interface................................487
Scroll bars 494
Selecting drives.............................495
Selecting files..........................499-501
Selecting menus 497-498
Shortcut keys...........................511-512
Starting.. 25
Starting programs.................35-36, 506
Status bar......................................494
Switching between programs 510
Task Swapper....................509, 513-516
Text mode......................................522
Title bar 494
Using parameters......................525-527
Using the mouse......................487, 491
Video Mode....................................522
Viewing files.......................32-33, 232
MS DOS Shell Disk Utilities:
Backup Fixed Disk 532
Command Prompt 530
Disk Copy 532
Format..533
Quick Format..................................532
Restore Fixed Disk..........................532
Undelete..533
MS-DOS Shell File menu....................536
Associate..................................541-542
Change Attributes............................547
Copy....................................537, 545
Create Directory..............................547
Delete...........................537, 545-546
Deselect All...................................548
Exit 538, 548
Move...544
New..537
Open............................537, 539-540
Print 540
Properties.......................................538
Rename..547
Reorder..538
Run.....................................538, 540
Search..542
Select All 548
View File Contents 543
MS-DOS Shell Help menu 559
About Shell.....................................560
Commands......................................560
Index...560
Keyboard..560
Procedures 560
Shell Basics....................................560
Using Help 560
MS-DOS Shell Options menu...............548
Colors..554
Confirmation...................................549
Display...553
Enable Task Swapper........................553
File Display Options.................550-551
Select Across Directories..................551
Show Information.....................551-553
MS-DOS Shell Tree menu...................558
Collapse Branch...............................559
Expand All......................................559
Expand Branch.................................558
Expand One Level............................558
MS-DOS Shell View menu............534-555
All Files...556
Dual File Lists 556
Program List 557
Program/File Lists............................557
Refresh..557
Repaint Screen................................557
Single File List................................555
NEAT chip set..........................391, 397
NOEMS..401
NOT..470
NOUMB...402
NUL.....................................631, 646
Number of entries 179
Numeric variables.............................584

O-P

Operating system 43, 279-280
Output device:
Default..328

- Defining 82

Output:
- Redirect to printer 331
- Redirecting 216-217, 328

Overlay files 291
OVL file 291
Parallel port 359
Parameters 9, 84, 476, 518, 525
Parity 362
PARK 64
Partition 111-113, 268
Partition table, protecting 268
PARTNSAV.FIL 269
Passwords 529
- In MS-DOS Shell 529-533

PATH 286, 353, 602, 610, 651
Path:
- Path specification 11, 177, 628
- Creating 185
- Deleting 190

PAUSE 475, 635
PCTRACKR.DEL 269
Pipe symbol "|" 319-321, 333, 482
Prevent Program Switch 523
Primary DOS partition 112
PRINT 50, 56, 458
Printer:
- *See also Printing*
- Connecting 233, 331
- Drivers 359
- Installing 359-361
- Output 359-360
- Printing 204
- Redirecting output 360-361
- Serial ports 362-363
- Using MODE 360-361

PRINTER.SYS 799
Printing:
- *See also Printer*
- Background 233
- Batch files 458
- Buffer size 234
- Connecting printer 233
- Directory 331-332
- Directory structure 204
- Files 540
- Graphics screen 54
- MS-DOS Shell 540
- Print queue 233-235
- Screen output 331

Program files 46, 285
Program Group (MS-DOS Shell) 516
Program Item (MS-DOS Shell) 516
Programs:
- *See also Application programs*
- Installing 281-282
- Starting 290
- Using installation programs 281

PROMPT 341-348
- Default PG 342
- In command interpreter 347-348
- In displaying directory 131-132, 184
- In environment variables 354
- Screen display 366-367
- Special characters 341
- Tips in using 609-610

Protected mode 388

Q-R

QBasic 532, 561-599
- Features 562-563
- Graphics command 596
- Line number 563
- Menus 568
- QBasic Editor 565
- QBasic pull-down menu 566
- QBasic scroll bar 566
- Quitting 582
- Saving programs 580
- Starting 563
- Startup screen 564
- Working in 561-562

QBasic Debug menu 574
- Clear All Breakpoints 574
- Procedure Step 574
- Set Next Statement 574
- Step 574
- Toggle Breakpoint 574
- Trace On 574

QBasic Edit menu 569-570
- Clear 570
- Copy 569
- Cut 569
- New FUNCTION... 570
- New SUB... 570
- Paste 570

QBasic File menu 569
- Exit 569
- New 569
- Open 569
- Print 569
- Save 569
- Save As 569

QBasic Help menu 577-578
- About 578
- Contents 578
- Index 577

- Topic ... 578
- Using Help ... 578-579

QBasic Options menu ... 575-576
- Display ... 575
- Help Path ... 576
- Syntax Checking ... 576

QBasic Run menu ... 573
- Continue ... 573
- Restart ... 573
- Start ... 573

QBasic Search menu ... 571-572
- Change... ... 572
- Find... ... 571
- Repeat Last Find ... 572

QBasic View menu ... 570
- Output Screen ... 570
- Split ... 570
- SUBs... ... 570

RAM ... 59, 381-382, 395

RAM disk ... 58, 389
- Advantages ... 152-153
- Alternatives ... 154
- Creating ... 152-156
- Creating with defaults ... 154-155
- Disadvantages ... 153-154
- Extended memory ... 155-156
- Removing ... 156
- Special defaults ... 155-156

RAMDRIVE.SYS ... 58-60, 152, 800

RD ... 207

Read/write slot ... 104

README text files ... 284

Ready prompt ... 53

Real mode ... 388

RECOVER ... 215, 274

Redirect:
- Creating pseudo drive ... 146-147
- Drive access ... 146-151
- Input ... 317, 327
- Output ... 328-329

Redirect characters:
- < ... 317, 327
- > ... 328
- >> ... 328-329

REN ... 238

RENAME ... 205, 411, 613

Replaceable parameter ... 518

Repositioning a group line ... 538

Repositioning a program item ... 538

RESTORE ... 239-240, 253-256, 263, 532
- Changing archive flag ... 241
- Changing system attribute ... 242-243
- Hard drive backups ... 253-256
- Hidden attributes ... 241-242
- Marking files ... 240
- Restoring all files ... 255-256
- Restoring deleted files ... 275-277
- Restoring selected files ... 263-264

Retry ... 363

Return value ... 465

ROM ... 381, 382

Root directory ... 173-174
- *See also Directory*
- Changing to root directory ... 183
- Defined ... 46
- Display all directories ... 201
- Display contents ... 184, 196
- Display hidden files ... 194
- Display files in subdirectories ... 199
- Illustrated ... 47

RUNDIR ... 826

S

SAA ... 421, 487

SAVER ... 826

Screen output:
- Background color ... 368
- Character display ... 366-367
- Clearing ... 316-317
- Displaying data ... 317-318
- Font color ... 367
- Setting ... 315-316

Screen Saver ... 826

Search path ... 286, 291
- Common search path ... 192
- Creating for files ... 190-191
- Deleting ... 190
- Displaying ... 186
- For programs ... 185-186
- Including directories ... 189-190
- Length ... 188
- Sequence ... 187

Search
- Contents of files ... 248
- For files ... 247
- Special search options ... 248

Secondary command interpreter ... 349

Sectors ... 161

Selecting files, MS-DOS Shell ... 499

Serial port ... 360

SET ... 352, 611

SETFUNC ... 831

SETUP ... 51, 215, 282

SHELL ... 345, 351, 601, 610, 625 625

SHIFT ... 476

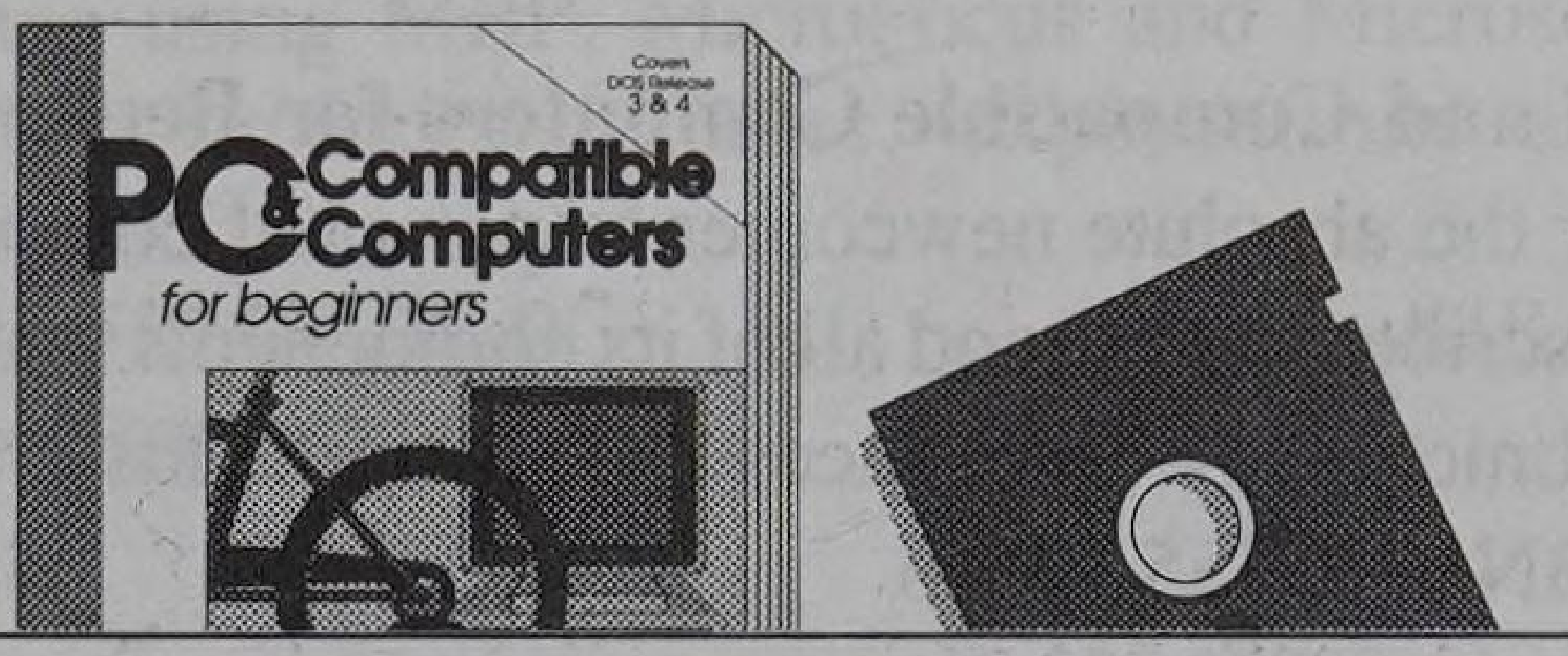
SIZE ... 832

SFORMAT ... 833

SMARTDRV.SYS..............58-60, 389, 801
SORT..............................315, 320, 327
Sort
  Direction....................................322
  nth column..................................322
  Options.......................................322
Source diskette...............................136
Source file.....................................218
Special characters 84
SPRINT...834
Standard output device.......................217
Starting a program 539
Startup file....................................... 36
Stop bits 363
Storage devices, checking...................141
Storage media 7, 245
  See also diskette
  Defined 48
  Deleting name...............................129
  Determining volume label.................130
  Displaying contents........................195
  Labeling.......................................129
Subdirectories....................8, 27, 196, 615
  Copying..224
  Directory names.......................175-177
  Displaying....................................649
  Extensions....................................613
  Number possible............................179
  Using directory names...............177-178
SUBST......................150, 207-209, 650
Substitute characters..........................198
System attribute 242
  Changing 242
  Clearing.......................................242
System diskette, formatting..........170, 171
System environment............................58
System files 126
System prompt..................2, 53, 67, 184
  See also PROMPT
  Customizing...........................341-342
  Using in CONFIG.SYS 342-344
System setting..................................352
System time 341
System variable..................348, 352, 610
  COMSPEC 610
  PATH...610
  PROMPT 610

T-X

Target directory................................218
Target diskette 136
TCD 835
TEMP 348, 354
Temporary file.............316, 319, 321, 515
Text:
  Creating files 215-216
  Finding specific text 323
  Finding strings...............................323
  Search options...............................324
  Searching......................................324
TIME.........................5-6, 309, 335-340
Trace mode......................................636
Tracks 161
TREE...................................12, 133-135
Truename.................................210, 775
TSR programs....................312-313, 508
TYPE............231, 320, 407-410, 458, 607
  Displaying ASCII files.................15-16
UMB 402
UNDELETE 21, 237, 276
  Undeleting files 276-277
UNFORMAT.............................270-271
  Switches 271
  System area....................................270
  Without backed up data...................271
Unformatting 619
Unused storage area...........................194
Upper memory.......................61, 386-398
  Layout 398
  Requirements..........................397-398
  Upper memory blocks (UMB) 55
  Using...397
Utilities.......................................39-40
Variable..480
Variable name..................................584
VDISK.SYS......................................152
VERIFY 355
VGA...364
VOL 130
Volume label.........................7, 170, 193
Volume label, assigning..............129, 130
Volume serial number 128
Wildcards 19, 198, 223, 238-246
  Character * 84-86, 213
  Character ?...........................84-86, 213
  Character symbols 84
  Functions in DOS 85
  Using in filenames...........................240
  Using to delete...............................236
  Using TYPE 607
  Using with DIR.........85-86, 133,198-199
Word processor..................................611
Write access, Verifying.......................355
Write protect flag 239
XCOPY 222, 239-240
XMS driver 401
XMS Memory 399


Covers DOS Release 3 & 4
PC & Compatible Computers
for beginners

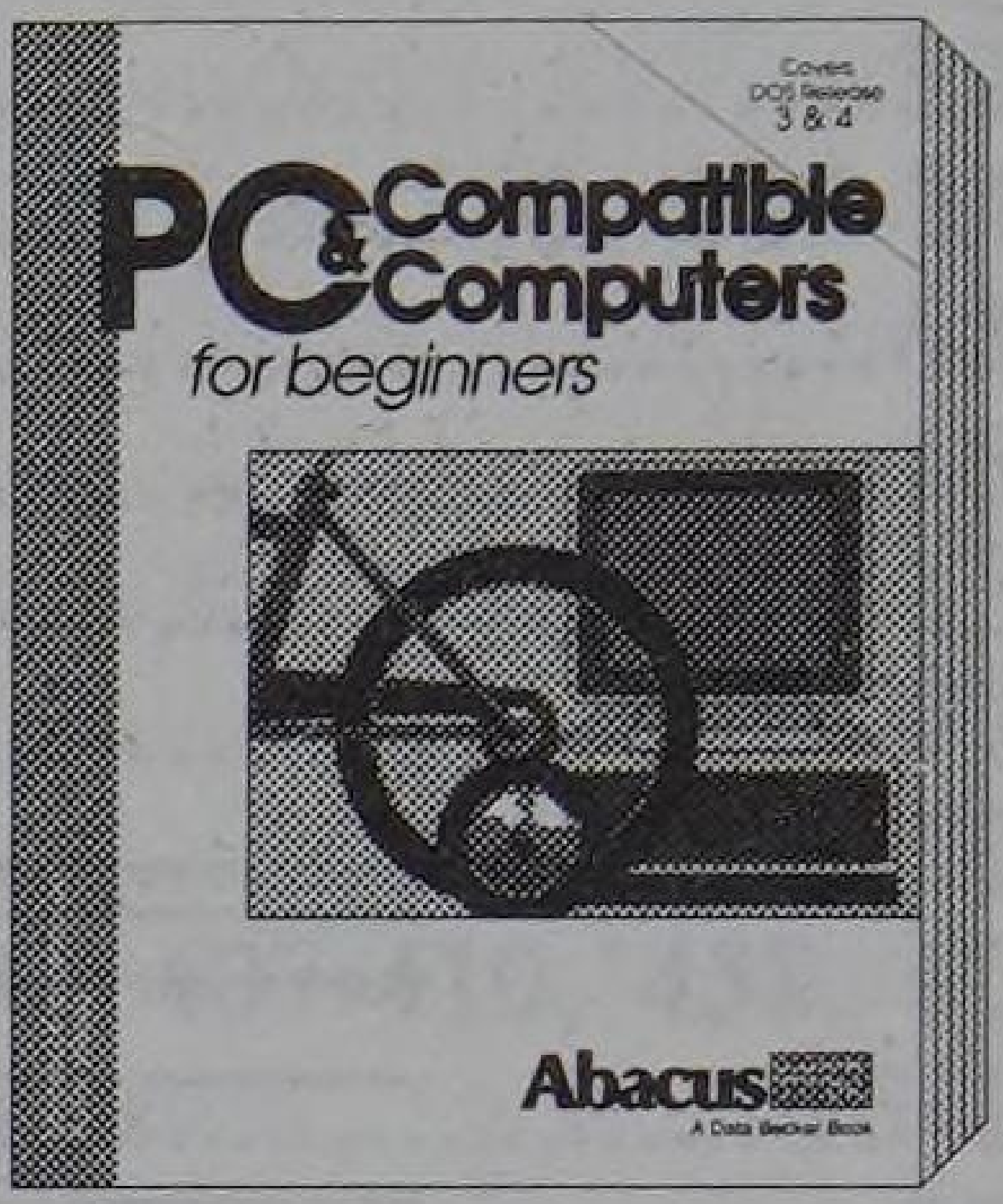
Abacus pc catalog

Order Toll Free 1-800-451-4319

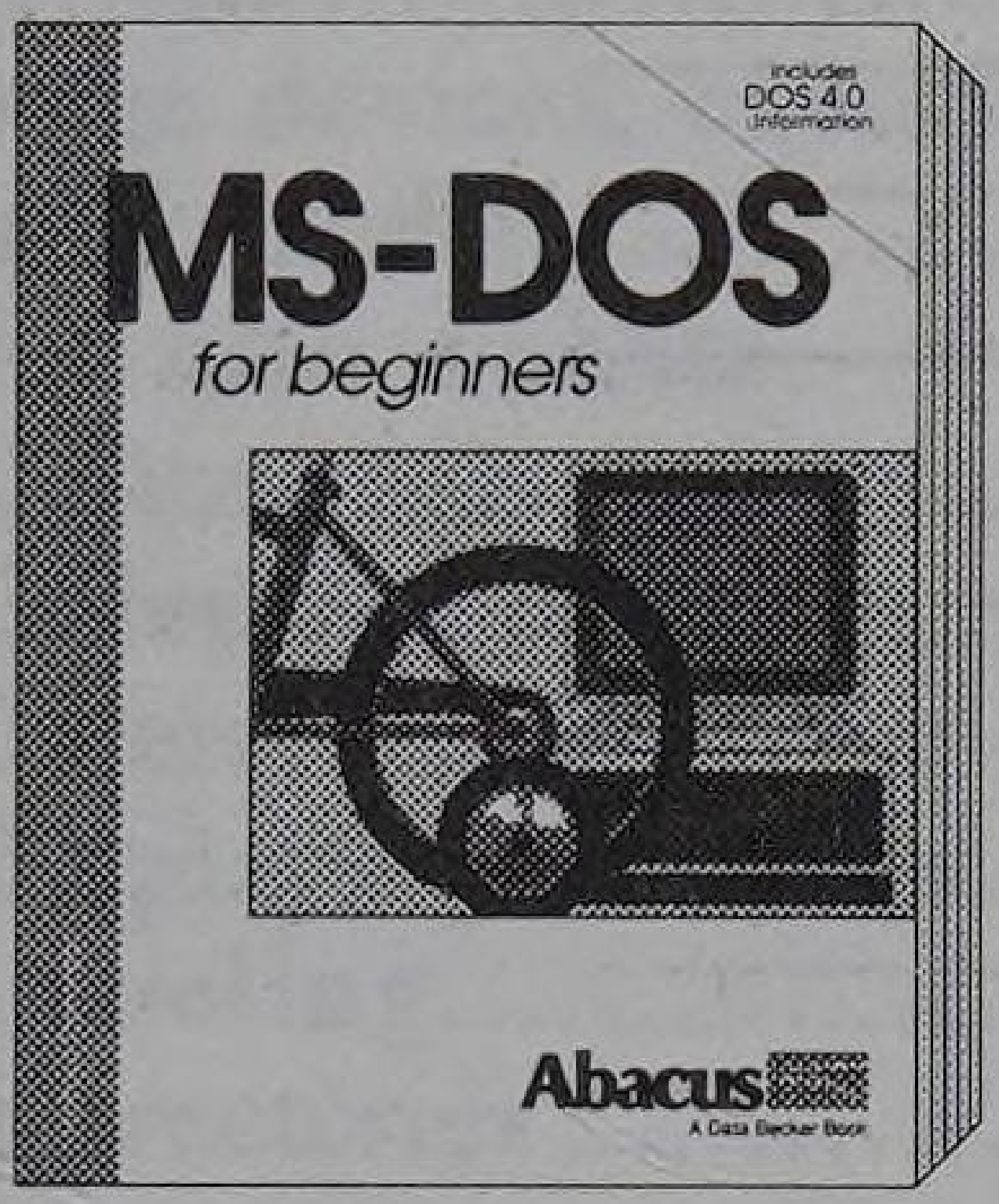
5370 52nd Street SE • Grand Rapids, MI 49512
Phone: (616) 698-0330 • Fax: (616) 698-0325

Beginners Books for new PC Users

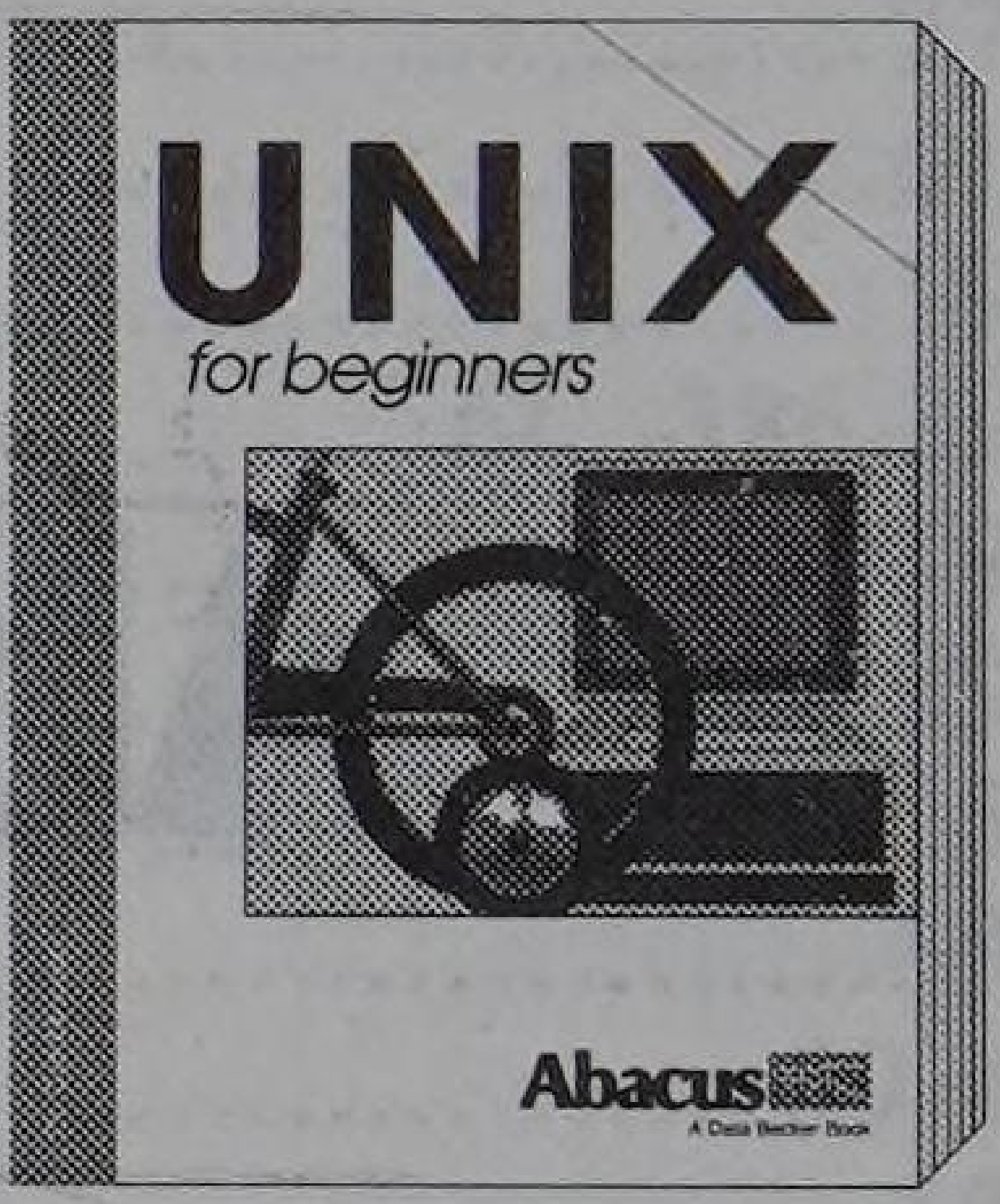
Beginners Series books remove the confusing jargon and get you up and running quickly with your PC.


Covers DOS Releases 3 & 4
PC & Compatible Computers
for beginners
Abacus
A Data Becker Book

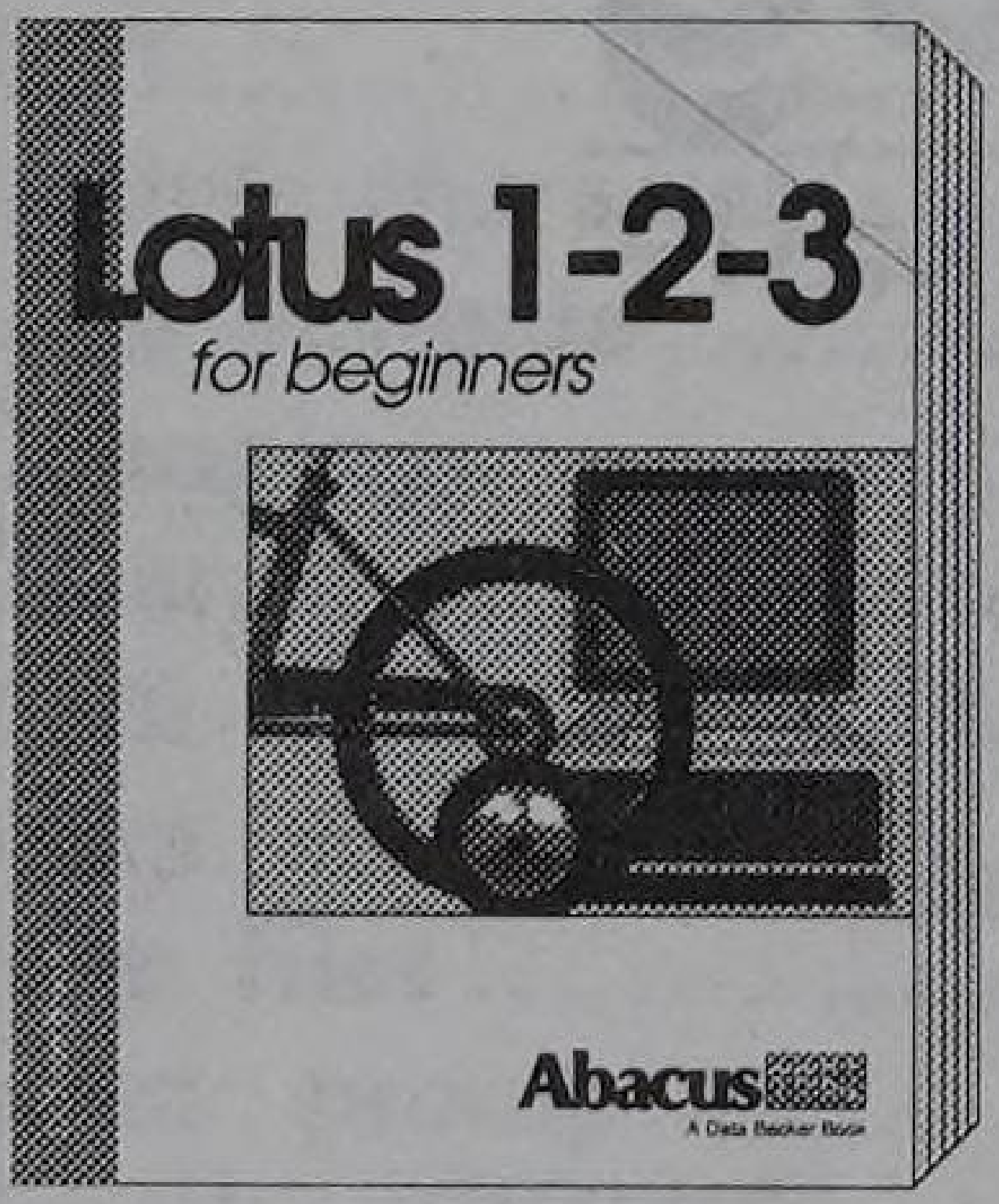
PC and Compatible Computers for Beginners - For the absolute newcomer to personal computers. Describes the PC and all of its components in a non-technical way. Introduces DOS commands.
ISBN 1-55755-060-3 $18.95
Canada: 52072 $22.95


Includes DOS 4.0 Information
MS-DOS
for beginners
Abacus
A Data Becker Book

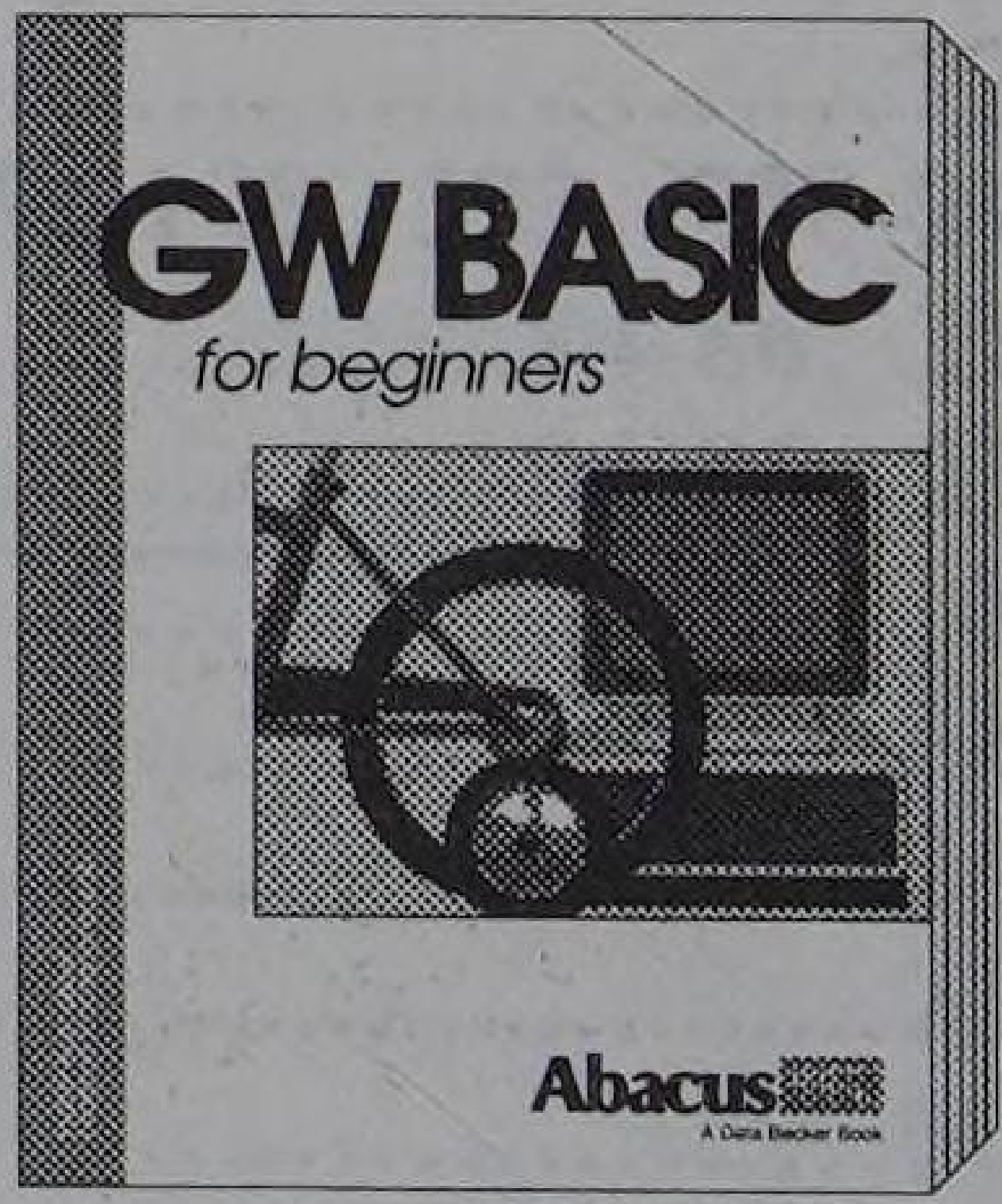
MS-DOS for Beginners - Describes the most important DOS commands clearly and understandably. Teaches skills required to use your PC more effectively.
ISBN 1-55755-061-1 $18.95
Canada: 52071 $22.95


UNIX
for beginners
Abacus
A Data Becker Book

UNIX for Beginners - Clearly describes this popular operating system, Logon procedures, file concepts and commands using simple and clear examples.
ISBN 1-55755-065-4 $18.95
Canada: 52073 $22.95


Lotus 1-2-3
for beginners
Abacus
A Data Becker Book

Lotus 1-2-3 for Beginners - Presents the basics with examples, clearly, without using confusing 'computer jargon'. Includes Release 2.2 information.
ISBN 1-55755-066-2 $18.95
Canada: 52069 $22.9


GW BASIC
for beginners
Abacus
A Data Becker Book

GW-BASIC Programming for Beginners* - A simple introduction to programming the PC using the BASIC language. Learn many of the commands writing sample programs and taking chapter quizzes.
ISBN 1-55755-062-X $18.95
Canada: 52068 $22.95

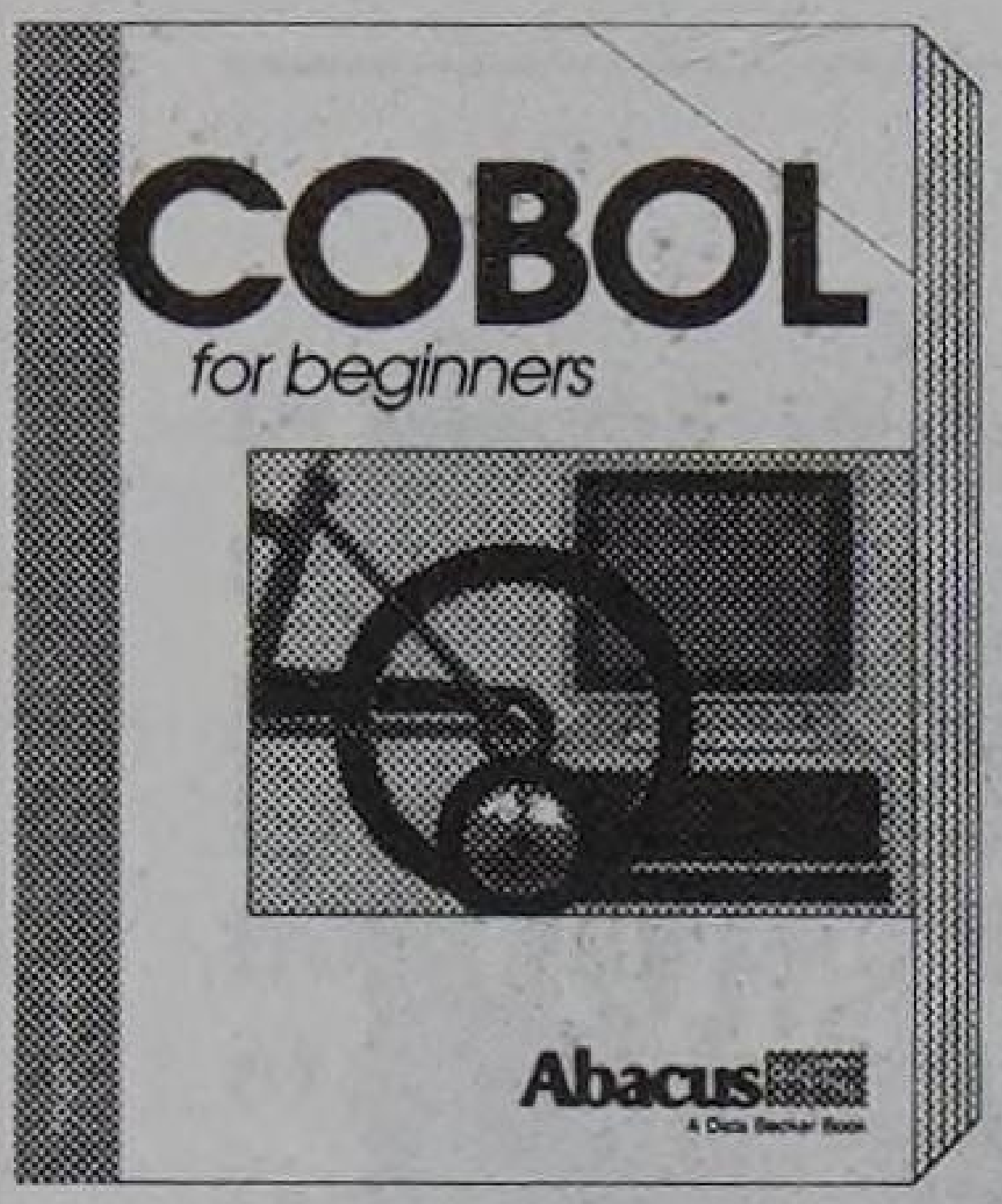
*Companion Disk available for $14.95 each ($19.95 CDN)

To order direct call Toll Free 1-800-451-4319

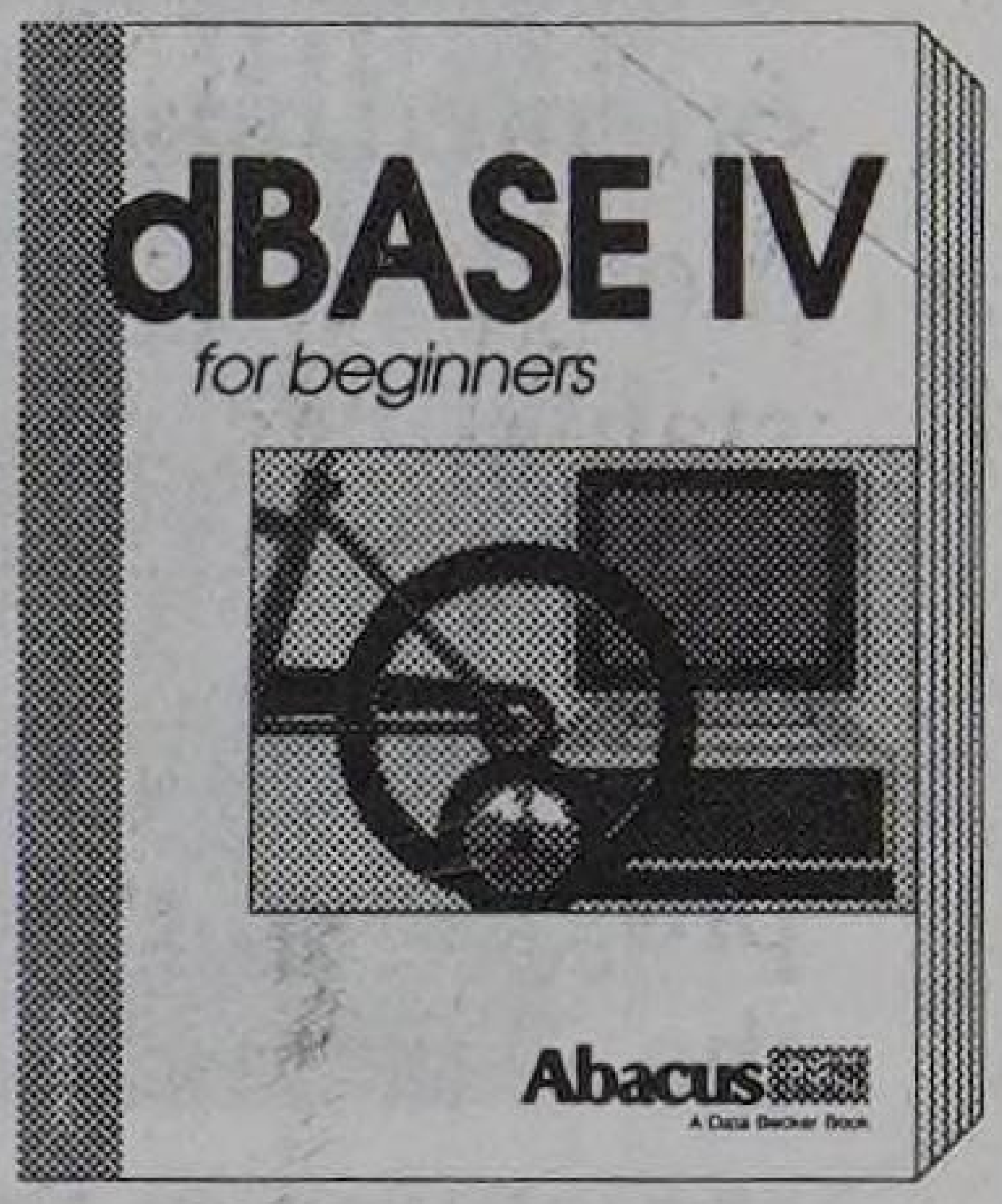
In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Beginners Books for new PC Users

Beginners Series books remove the confusing jargon and get you up and running quickly with your PC.


COBOL
for beginners
Abacus
A Data Becker Book

COBOL for Beginners* - Teaches this popular language using MBP, MicroFocus and Microsoft COBOL. Includes quizzes, explanations and demonstrations.
ISBN 1-55755-070-0 $18.95
Canada: 53904 $22.95


dBASE IV
for beginners
Abacus
A Data Becker Book

dBASE IV for Beginners - Simply and easily explains the fundamentals of dBASE. Learn to operate this package quickly and use its powerful commands and functions.
ISBN 1-55755-069-7 $18.95
Canada: 52066 $22.95

NEW
QuickBASIC
for beginners
Abacus
A Data Becker Book

QuickBASIC for Beginners - Learn QuickBASIC quickly and easily. Begins by teaching you to write small and simple applications to simplify your everyday tasks.Complete descriptions of all QuickBASIC commands.
ISBN 1-55755-111-1. $18.95
Canada: $22.95

NEW
DOS 5.0
for beginners
Abacus
A Data Becker Book

DOS 5.0 for Beginners - Gets the new user up and running quickly using Microsoft's latest DOS 5.0. You'll find clear explanations and plenty of "hands on" examples here. Describes the most important DOS commands clearly and understandably.
ISBN 1-55755-112-X. $18.95
Canada: $22.95

NEW
Word Perfect
for beginners
Abacus
A Data Becker Book

WordPerfect for Beginners - Contains all the important information about WordPerfect.Learn the essentials of WordPerfect's command set through practical examples - from writing a letter to composing longer documents.
ISBN 1-55755-117-0. $18.95
Canada: $22.95

*Companion Disk available for $14.95 each ($19.95 CDN)

To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Developers Series books are for the professional software developer who requires in-depth technical information and programming techniques.

Turbo Vision Toolbox

NEW
Developer's Series
TURBO VISION
TOOLBOX
An in-depth reference to object oriented programming
Michael Tischer
Includes ready-to-use companion diskette
Abacus
A Data Becker Book

is the reference manual that teaches you step-by-step to program in Borland's popular OOP (object oriented programming) Turbo Vision. You will learn how to program menus, status lines, windows and dialog boxes. **Turbo Vision Toolbox** also includes a complete editor that .will get you started programming quickly. You'll find all the Turbo Vision objects and data structures in the guide as well.

This toolbox enables you to "construct" the program interface of your application into the dialog, which automatically creates the Turbo Pascal source code. You only need add your functions and procedures and your program is ready to run.

Beginners can create complex Turbo Vision programs easily without having to worry about objects and pointers. **Turbo Vision Toolbox** contains Units for those who like to work without OOP and pointers. **Turbo Vision Toolbox** also contains a number of dialog boxes that can be added to the programs.

Turbo Vision Toolbox introduces you to Turbo Vision with installation tips and advice about the Toolbox. All of the elements such as input lines, radio buttons and check boxes, buttons, lists and scroll bars are explained. You will also find many examples, and a large reference section covering Turbo Vision, the Toolbox and the Toolbox Units.

ISBN 1-55755-147-2
Suggested retail price $39.95.
Available February 1992

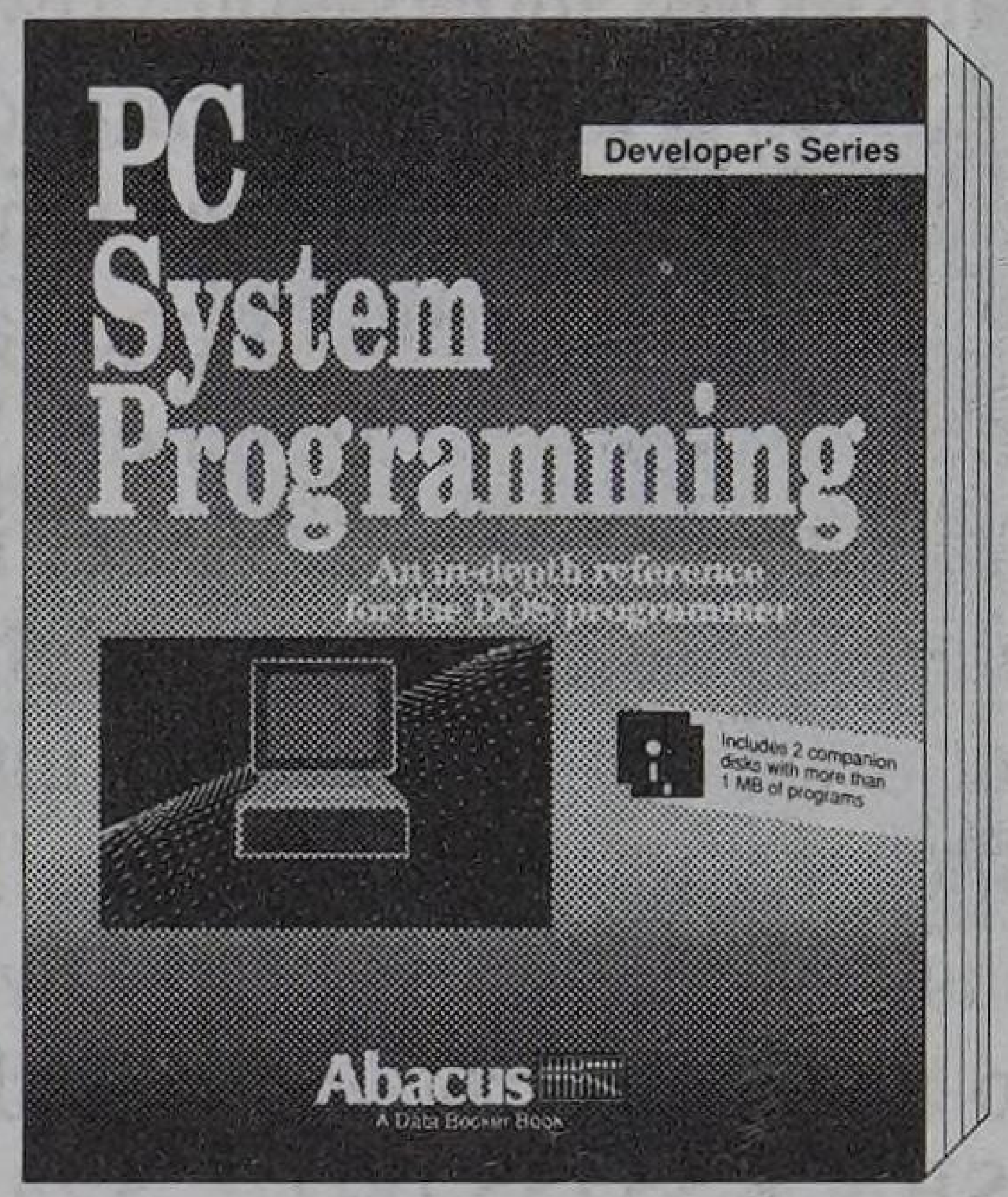
Includes ready-to-use Companion Diskette

To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Developer's Series Books

Developers Series books are for the professional software developer that requires in-depth technical information and programming techniques.


PC System Programming
Developer's Series
An in-depth reference for the DOS programmer
Includes 2 companion disks with more than 1 MB of programs
Abacus
A Data Becker Book

PC System Programming is a literal encyclopedia of technical and programming knowledge. Features parallel working examples in Pascal, C, ML and BASIC. Explains memory layout, DOS operations and interrupts from ML and high level languages, using extended, expanded memory, device drivers, hard disks, PC ports, mouse driver programming, fundamentals of BIOS, graphics and sound programming, TSR programs and complete appendices. 930 page book and two disks with over 1MB of source code.
ISBN 1-55755-036-0. $59.95
Canada: 52444 $74.95

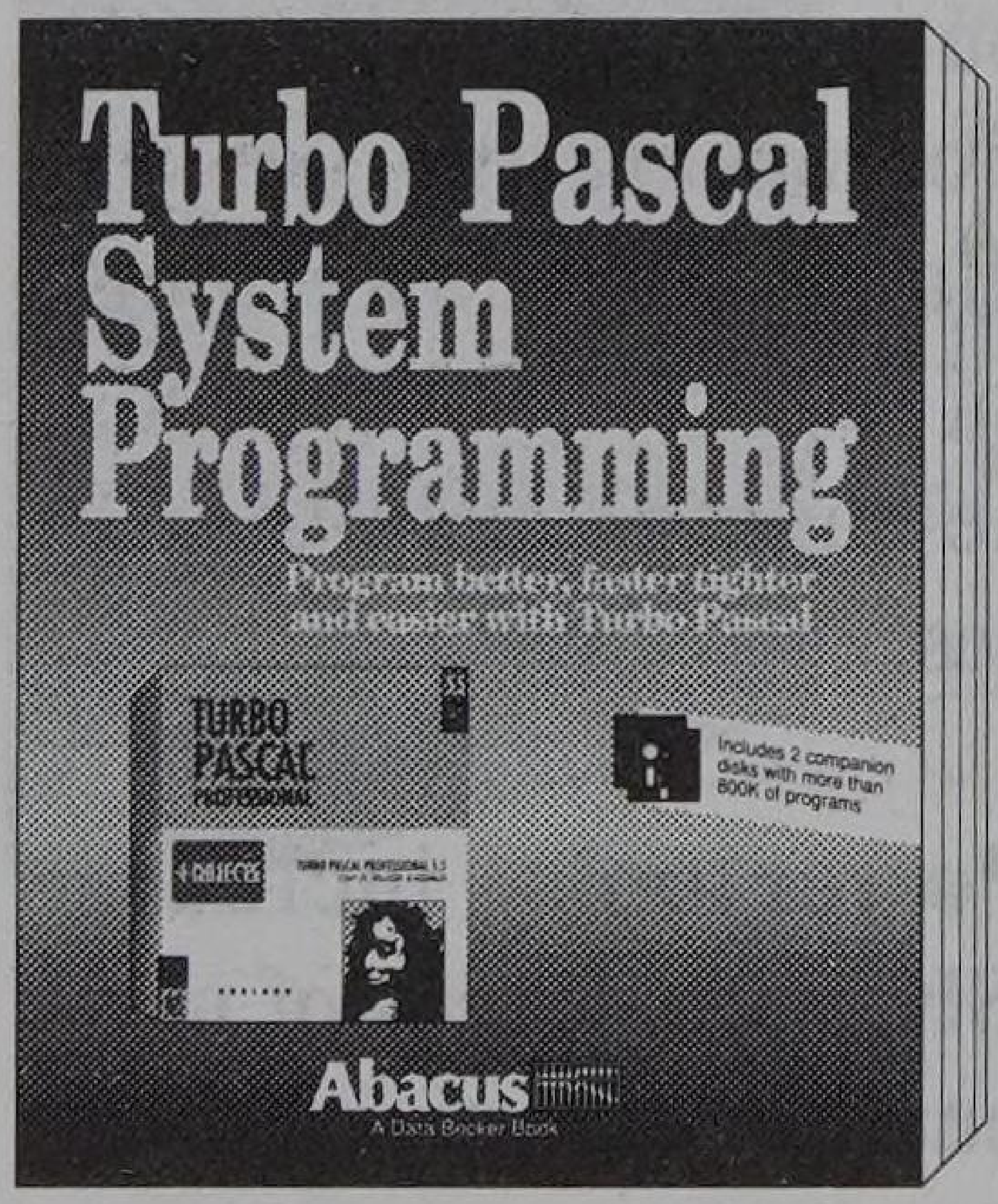
What readers are saying about
PC System Programming:

"The day I saw this book I bought it" F.R.

"Excellent work" G.O.

"This book is such a good reference and so complete and in-depth, I keep it next to my system reference manuals and my compiler manuals" J.K.

"Best written PC language book I've seen" AVR, Tulsa, OK.


Turbo Pascal System Programming
Program better, faster tighter and easier with Turbo Pascal
Includes 2 companion disks with more than 800K of programs
TURBO PASCAL PROFESSIONAL
Abacus
A Data Becker Book

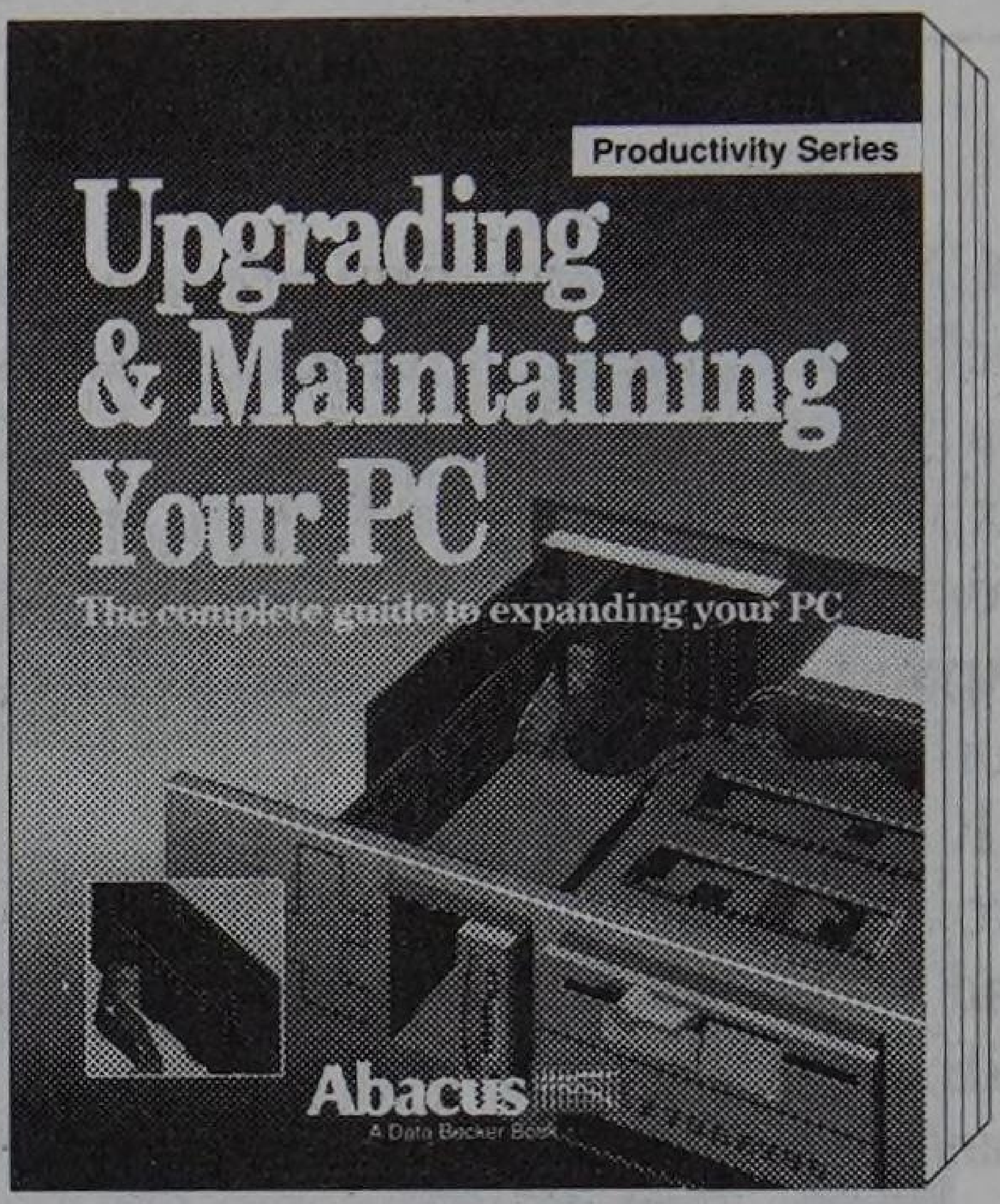
Turbo Pascal System Programming gives you the *know how* to write better, faster, tighter and easier code using the popular Turbo Pascal system. Author Michael Tischer has written this book for other pros who need to understand the ins and outs of using Turbo for their programming projects.

Turbo Pascal System Programming details dozens of important subjects including writing Terminate and Stay Resident (TSR) programs using Turbo; building windows and menus; using expanded and extended memory; writing multi-tasking programs; understanding Turbo's memory management; combining assembly language with Turbo programs; using interrupts for system programming tasks; accessing the hardware directly and more.
750 page book includes two disks with more than 800K of source code.
ISBN 1-55755-124-3. $44.95
Canada: 53910 $64.95

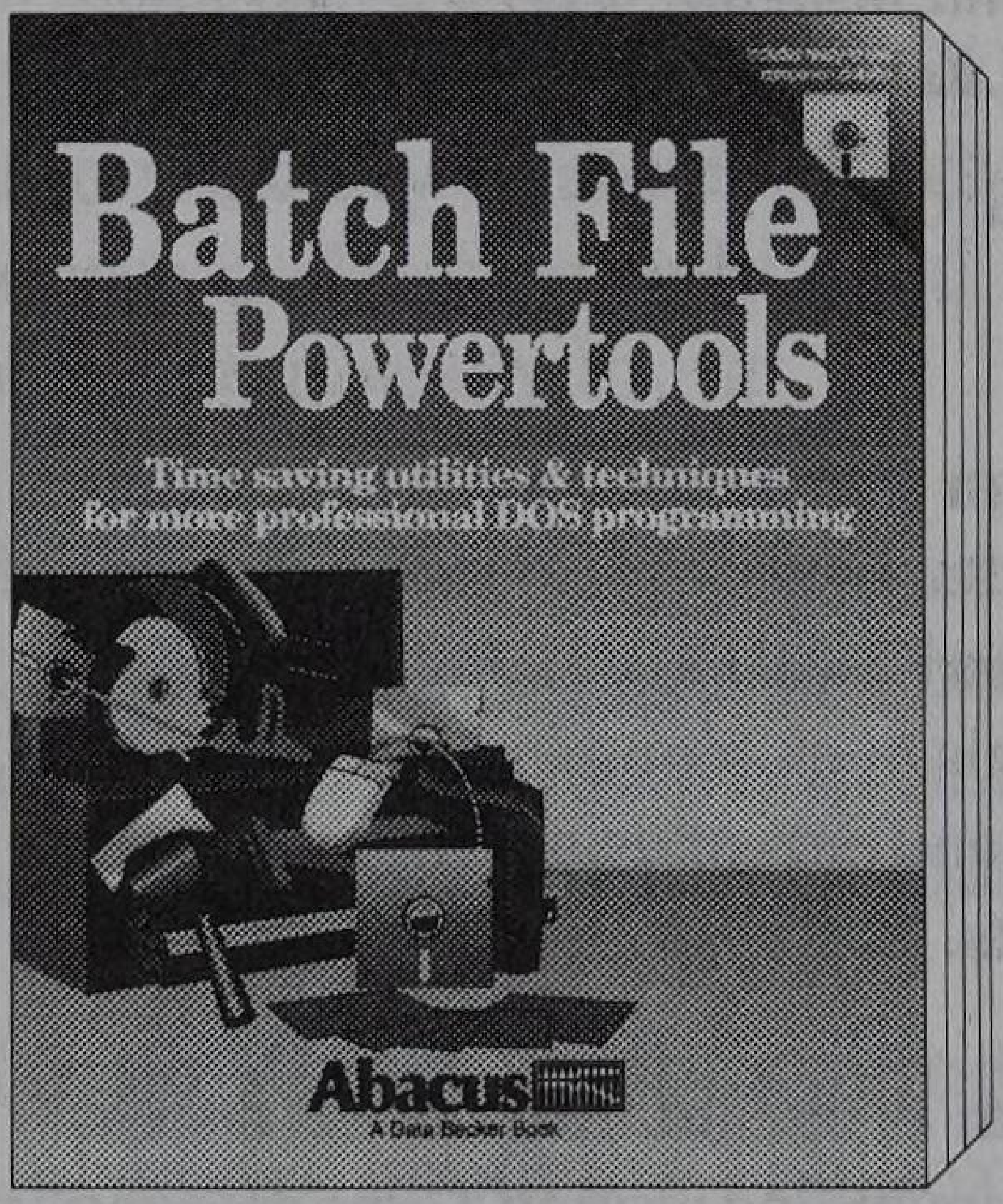
To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Productivity Series Books


Productivity Series
Upgrading & Maintaining Your PC
The complete guide to expanding your PC
Abacus
A Data Becker Book

Upgrading & Maintaining your PC
Your PC represents a major investment. This book shows you how to turn your PC into a high performance computing machine. It describes what you'll see when you open the "hood" and how all of the parts work together. Whether you want to add a hard drive, increase your memory, upgrade to a higher resolution monitor, or turn your XT into a fast AT or 386 screamer, you'll see how to do it easily and economically, without having to be an electronics wizard.
ISBN 1-55755-092-1. $24.95
Canada: 53926 $33.95


Batch File Powertools
Time saving utilities & techniques for more professional DOS programming
Abacus
A Data Becker Book

Batch files make your computer sessions easier. **Batch File Powertools** shows you how to use powerful, easy-to-learn techniques for many DOS applications. You'll boost your computing productivity with these techniques for making truly powerful batch files. **Batch File Powertools** includes dozens of new batch commands for writing time-saving, easy-to-use "power" batch files.

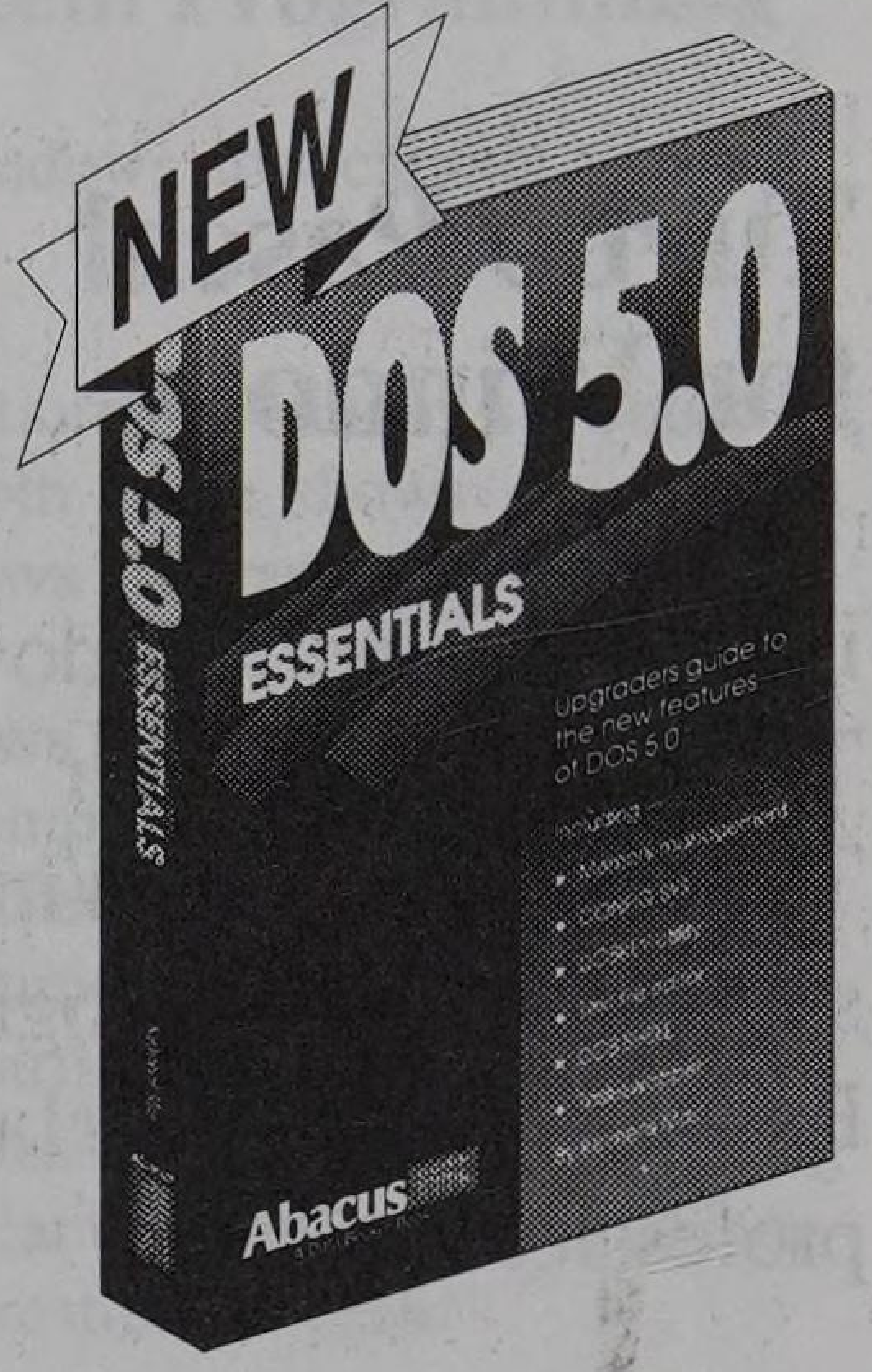
Batch File Powertools includes BatchBASIC on the companion disk. BatchBASIC adds dozens of new commands to DOS to help increase your computing productivity. The companion disk also contains dozens of practical examples.
ISBN 1-55755-102-2. $34.95
Canada: $45.95

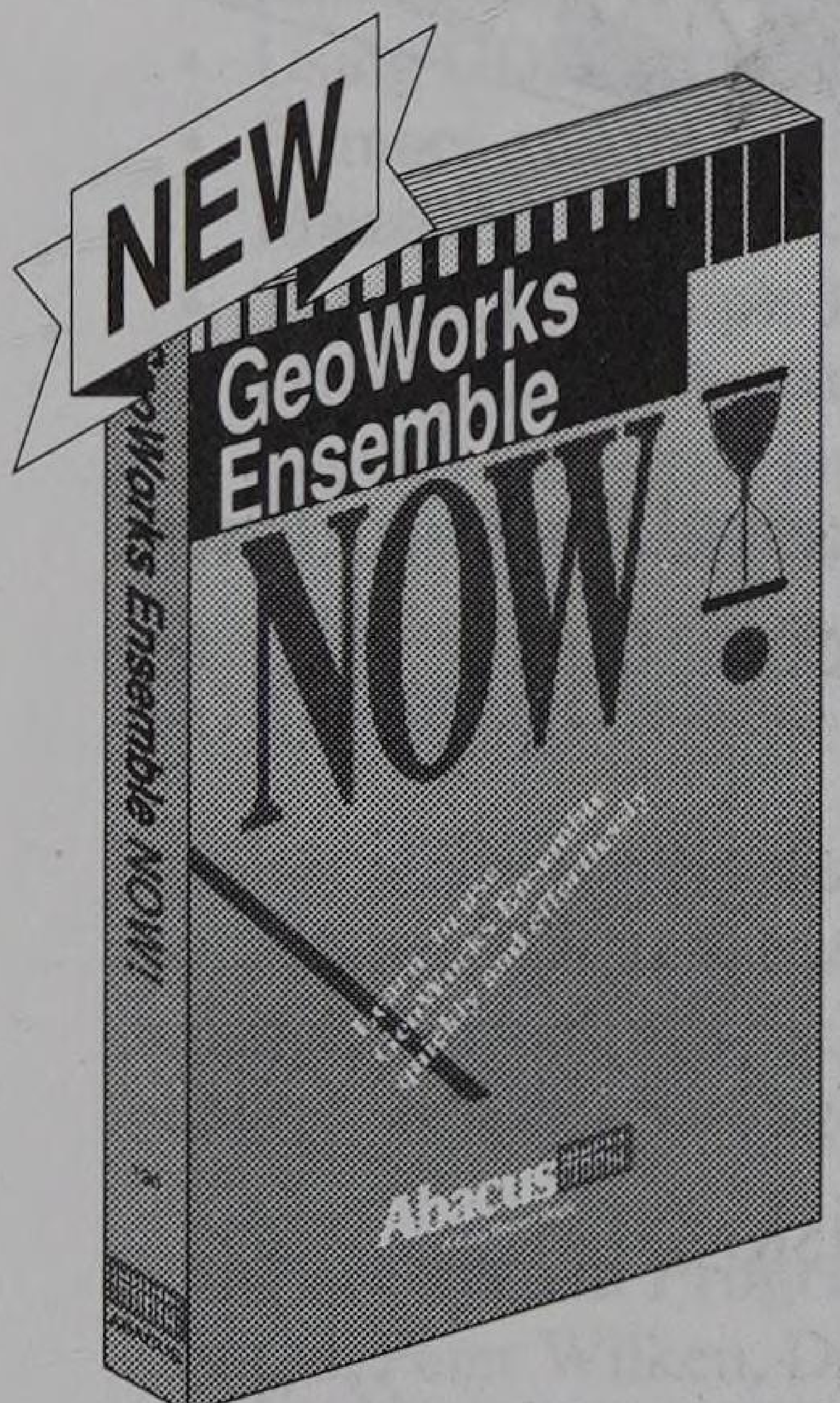
To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Productivity Series Books

DOS 5.0 Essentials
is the fast and effective way to learn the MS-DOS 5.0 operating system quickly and easily. Shows how you can operate DOS on a daily basis - from deleting and restoring files to generating macros to using commands and parameters. Using the menu-oriented DOS Shell makes operating system management even easier, right down to fast access to the DOS command line. Topics include: working with files, directories at a glance, storage media - diskettes and hard disks, using the DOS Shell and Program Manager, using the DOS command line, DOS Editor and more.
Author: Kelmens Mai
ISBN 1-55755-115-4. $14.95





GeoWorks Ensemble Now!
is your guide to the new graphic interface for PC XT, AT and 386 machines will make your computing life easier. **GeoWorks Ensemble Now!** is the fast and effective way to start using GeoWorks Ensemble immediately. This book features tutorials and concise summaries including: Window management, Applications in GeoWorks Ensemble, GeoManager, Data exchange in GeoWorks Ensemble, Using "normal" MS-DOS applications and much more.

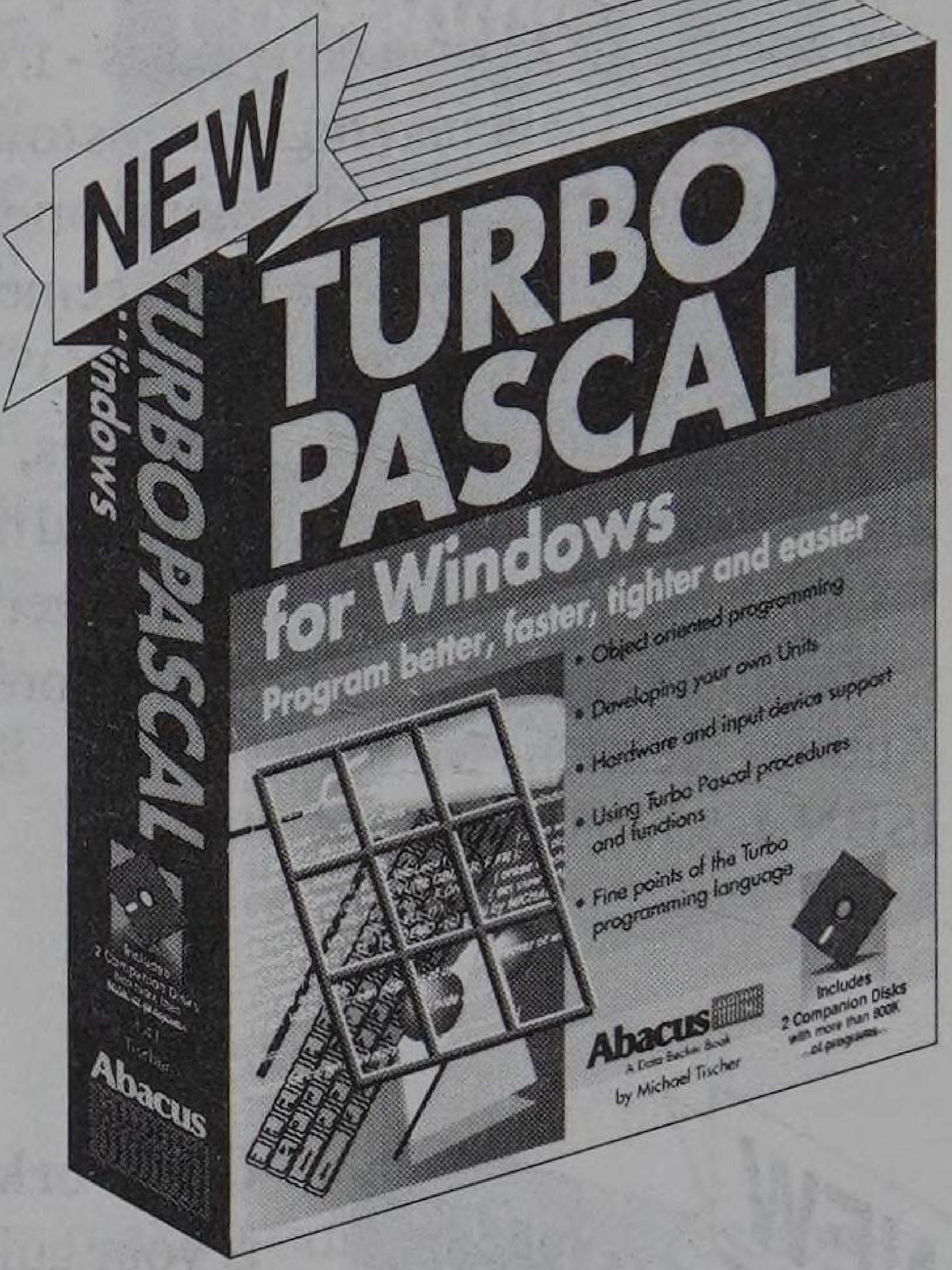
Including:
- Installation
- General Skills
- Applications in GeoWorks Ensemble
- GeoManager
- More about GeoWorks Ensemble

Available August.
Author: R. Albrecht and M. Plurg
ISBN 1-55755-120-0. $12.95

To order direct call Toll Free 1-800-451-4319
In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Developers Series books are for the professional software developer who requires in-depth technical information and programming techniques.

Turbo Pascal for Windows


NEW
TURBO PASCAL
for Windows
Program better, faster, tighter and easier
• Object oriented programming
• Developing your own Units
• Hardware and input device support
• Using Turbo Pascal procedures and functions
• Fine points of the Turbo programming language
Includes 2 Companion Disks
Abacus
A Data Becker Book
by Michael Tischer
TURBO PASCAL
for Windows
Abacus

is the complete 'Window' to using Turbo Pascal under Windows. **Turbo Pascal for Windows** will show you how to program Turbo Pascal from Borland with professional results.

Turbo Pascal for Windows includes an extensive introduction to the integrated Turbo Pascal environment. This guide includes a practical overview of the fundamentals and special features of Pascal programming under Windows. **Turbo Pascal for Windows** also includes an introduction to the Unit Concept that's vital to understanding object oriented programming.

Turbo Pascal for Windows. ISBN 1-55755-141-3.
Authors: Gunther Faeber and Martin Pauly.
Suggested retail price $39.95 with companion disk.
Available February 1992.

Includes ready-to-use Companion Diskette

To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

NEW
WINDOWS
SYSTEM
PROGRAMMING
The guide to programming professional Windows applications
By Peter Wilken and Dirk Honekamp
Includes ready-to-use companion disk
Abacus
A Data Becker Book

Windows System Programming

Learn to program Windows applications faster and easier.

Programming applications for Windows is said to be complex. This in-depth guide shows you how to speed up your Windows programming projects.

This bible for Windows programmers includes the tools you need to improve your programming techniques. Contains several sample applications with exhaustive documentation and background information, saving you time and effort in developing your own applications. Numerous programming examples in the popular C language lead you step-by-step through the creation process.

Topics include:

- Introduction to the Windows programming
- Using GDI for graphic and text output
- Bitmaps and the color palette manager
- Managing both static and dynamic dialog boxes
- The MDI (Multiple Document Interface)
- Data exchange between applications using DDE
- Parallel and serial interface access
- Windows memory management
- Fine tuning Windows applications
- The DLL (Dynamic Link Libraries) concept

Includes a companion diskette containing both source codes and EXE files of sample applications. The source codes can be compiled using Microsoft C Version 5.1 and higher (not included with this package) and the Microsoft Software Development Kit (SDK - not included with this package).

Windows System Programming.
Authors: Peter Wilken, Dirk Honekamp
Item #B116. ISBN 1-55755-116-2.
Suggested retail price $39.95 with companion disk.

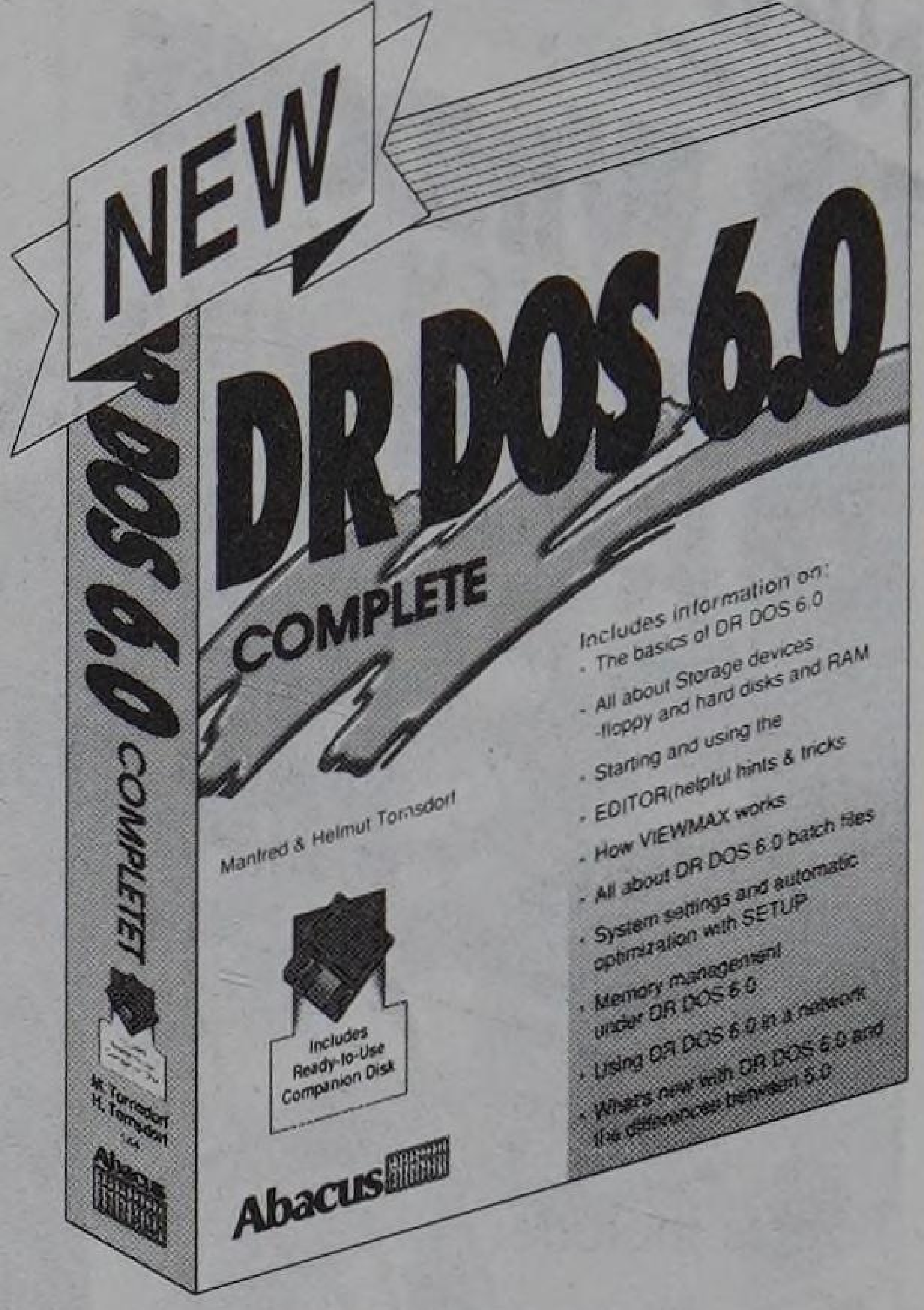
To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Productivity Series books are for users who want to become more productive with their PC

DR DOS 6.0 Complete

is the everyday user's guide to running Digital Research's brand new DR DOS 6.0 operating system. This book shows you how to use DR DOS 6.0's new features including: SuperStor disk compression, MemMAX memory saving, FileLink data transfer and more. Both beginning and advanced users will find that the techniques presented will make them more productive with DR DOS 6.0.

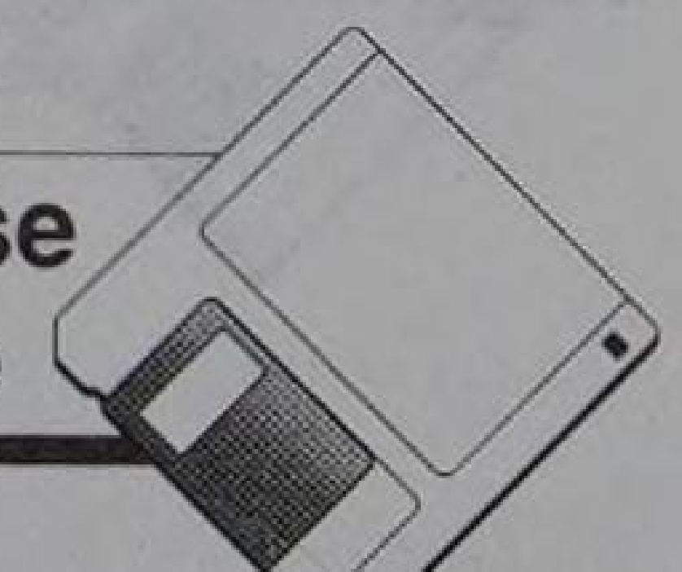

NEW
DR DOS 6.0
COMPLETE
Manfred & Helmut Tornsdorf
Includes information on:
· The basics of DR DOS 6.0
· All about Storage devices -floppy and hard disks and RAM
· Starting and using the EDITOR(helpful hints & tricks)
· How VIEWMAX works
· All about DR DOS 6.0 batch files
· System settings and automatic optimization with SETUP
· Memory management under DR DOS 6.0
· Using DR DOS 6.0 in a network
· What's new with DR DOS 6.0 and the differences between 5.0
Includes Ready-to-Use Companion Disk
Abacus

Just a few of the topics include:

- An introduction to DR DOS 6.0
- Basics of DR DOS 6.0
- Understanding and using the SSTOR command
- Disk Management with DR DOS 6.0
- How DR DOS 6.0 handles directories & file management
- Using expanded memory for the OS
- Running with Windows
- Starting and using the EDITOR (helpful tips & tricks)
- Understanding and using the ViewMAX command
- All about DR DOS 6.0 batch files
- Using DR DOS 6.0 in a network
- Memory management with DR DOS 6.0
- Optimizing your hard drive
- Includes detailed DR DOS Command Reference table and much more

DR DOS 6.0 Complete. 750 pages. ISBN 1-55755-144-8.
Authors: Manfred Tornsdorf & Helmut Tornsdorf.
Suggested retail price $34.95 with companion disk.

Includes ready-to-use Companion Diskette



To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Productivity Series books are for users who want to become more productive with their PC.

Intro to Windows Programming

Intro to Windows Programming walks the programmer through the seemingly complex tasks of writing Windows applications. It introduces the reader to the overall concept of Windows programming and events using dozens of easy to follow examples. It's a complete introductory text for beginning or intermediate Windows programmers. Includes 3 1/2" companion diskette with example programs.

Just a few of the topics of **Intro to Windows Programming** include:

- The Windows message system
- Text and graphics output via Device Context
- Device dependent and device independent bitmaps
- File management under Windows
- The structure of the Windows help system
- The multiple document interface
- Creating and managing dialog boxes and their elements
- Keyboard, mouse and timer input
- Communication with the clipboard
- Dynamic Link Libraries and much more

ISBN 1-55755-139-1.Author(s): Juergen Baer, Irene Bauder.
Suggested retail price $34.95 with 3 1/2" companion disk.

Includes ready-to-use Companion Diskette

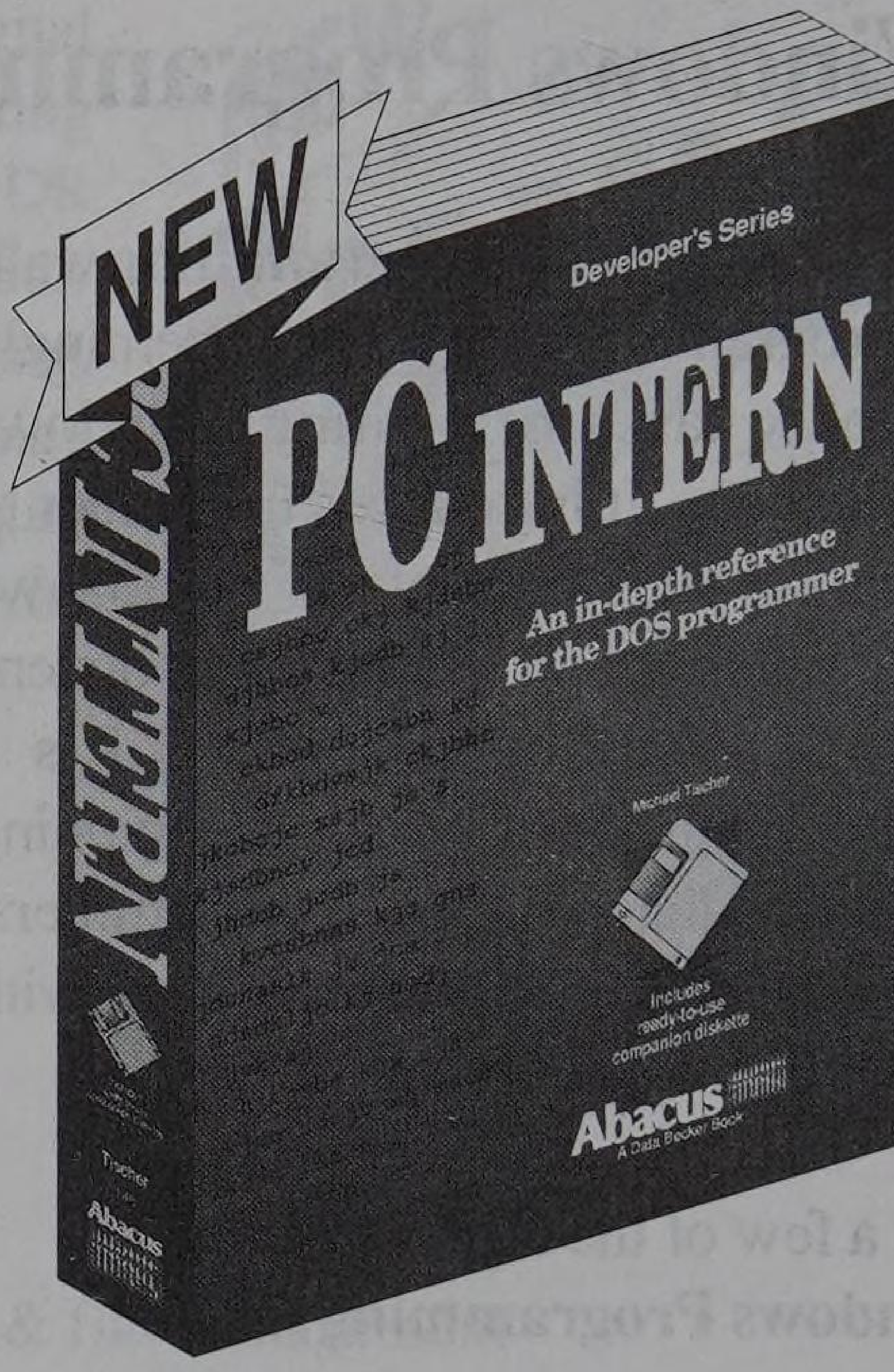
To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Developers Series books are for the professional software developer who requires in-depth technical information and programming techniques.

PC Intern

is a completely revised edition of our bestselling PC System Programming book which exceded 100,00 copies. **PC Intern** is a literal encyclopedia of for the DOS programmer. Whether you program in assembly language, C, Pascal or BASIC, you'll find dozens of practical, parallel working examples in each of these languages. **PC Intern** clearly describes the technical aspects of programming under DOS. More than 1000 pages are devoted to making DOS programming easier. Some of the topics covered include:


NEW
Developer's Series
PC INTERN
An in-depth reference for the DOS programmer
Michael Tischer
Includes ready-to-use companion diskette
Abacus
A Data Becker Book
PC INTERN
Tischer
Abacus

- PC memory organization
- Using extended and expanded memory
- Hardware and software interrupts
- COM and EXE programs
- Handling program interrupts in BASIC, Turbo Pascal, C and assembly language
- DOS structures and functions
- BIOS Fundamentals
- Programming video cards
- TSR programs
- Writing device drivers
- Multi-tasking

Includes ready-to-use Companion Diskette

The clearly documented examples make it easy for the reader to adapt the programs for his own requirements.

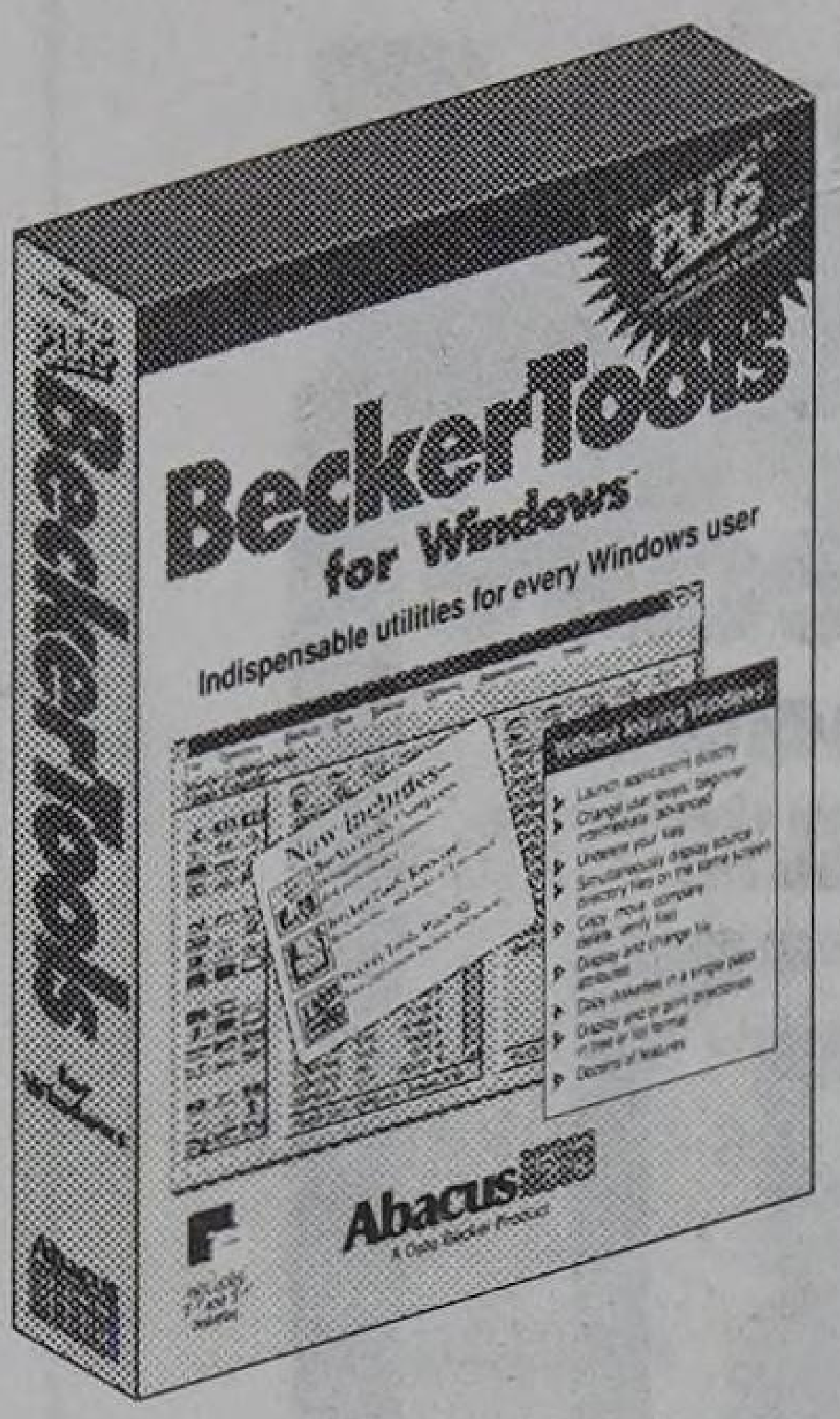
ISBN 1-55755-145-6. Author : Michael Tischer
Suggested retail price. $ 59.95. Available February 1992.

To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

New BeckerTools 2.0 PLUS for Windows:

Indispensable utilities for every Windows user



If you're a Windows user, you'll appreciate **BeckerTools Version 2** for Windows. **BeckerTools** will dramatically speed-up your file and data management chores and increase your productivity. Where Windows' File Manager functions end, **BeckerTools** features begin by giving you powerful, yet flexible features you need to get the job done quickly. **BeckerTools** has the same consistent user interface found in Windows so there's no need to learn 'foreign' commands or functions. Manipulating your disks and files are as easy as pointing and clicking with the mouse. You'll breeze through flexible file and data functions and features that are friendly enough for a PC novice yet powerful enough for the advanced user. You won't find yourself 'dropping out' of Windows to perform powerful, essential DOS functions like undeleting a file or waiting for a diskette to be formatted. **BeckerTools** has the enhanced applications available at the click of a mouse button. Item #S110 ISBN 1-55755-110-3. With 3 1/2" and 5 1/4" diskettes. Suggested retail price $129.95.

Now includes—

BeckerTools Compress
Defragments and optimizes disk performance.

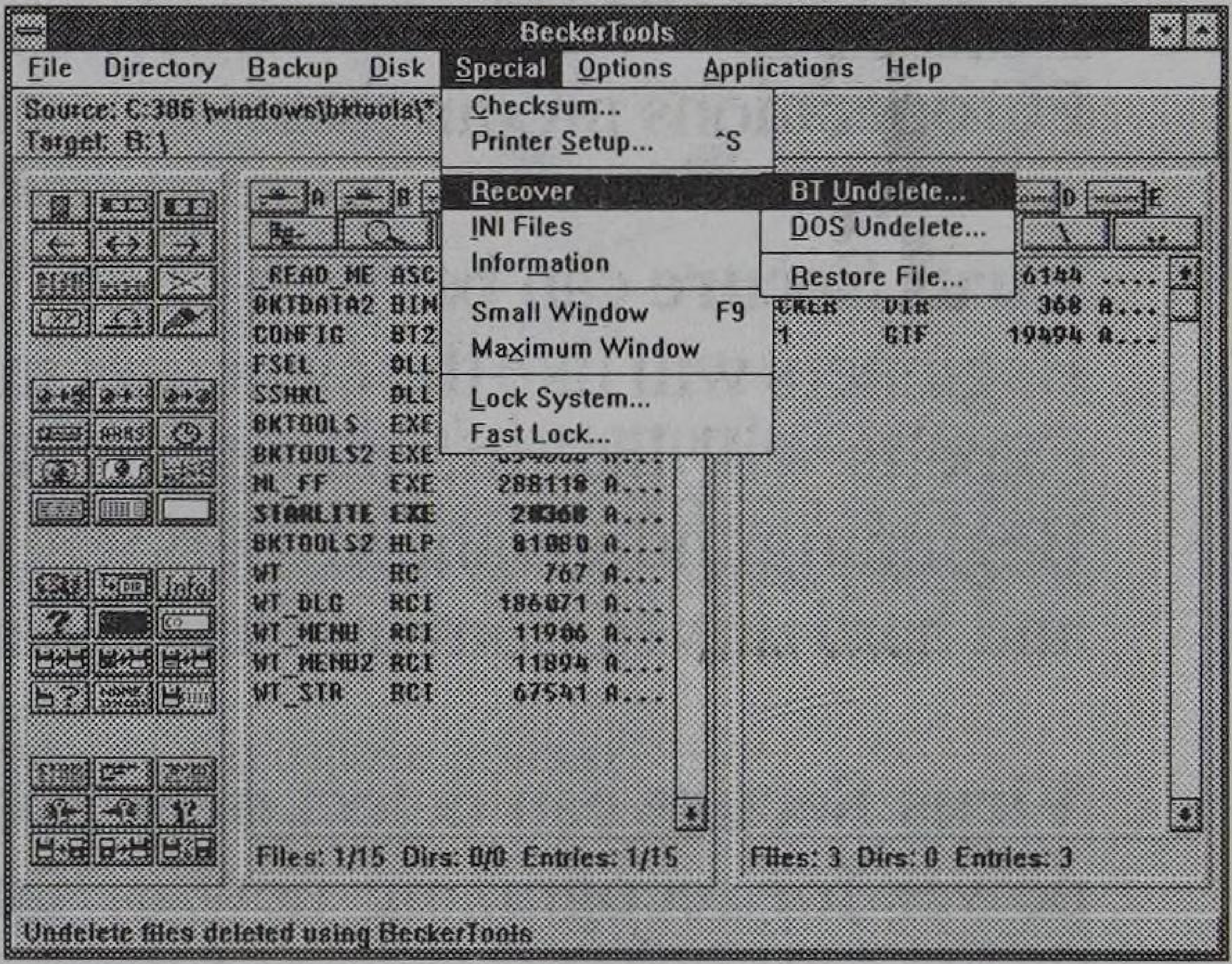
Optimizes your hard disk performance by defragmenting your files and/or disk. Multiple compress option lets you choose the type of optimization.

BeckerTools Recover
Rescues files and disks.

Checks and repairs common file and disk problems-corrupted FAT's, improperly chained clusters, bad files, etc. Could be a "Life-saver".

BeckerTools Backup
Fast, convenient file backup and restore.

File and disk backup/restore at the click of a button. Multiple select options, optional data compression, password protection and more.



```
Source: C:386 \windows\bktools\*.
Target: B:\

_READ_ME ASC
BKTDATA2 BIN
CONFIG   BT2
FSEL     DLL
SSHKL    DLL
BKTOOLS  EXE
BKTOOLS2 EXE
NL_FF    EXE     288118 A...
STARLITE EXE      28368 A...
BKTOOLS2 HLP      81080 A...
WT       RC         767 A...
WT_DLG   RCI     186071 A...
WT_MENU  RCI      11986 A...
WT_MENU2 RCI      11894 A...
WT_STR   RCI      67541 A...
Files: 1/15  Dirs: 0/0  Entries: 1/15

              6144 ....
         DIR   368 A...
1        GIF 19494 A...
Files: 3  Dirs: 0  Entries: 3

Undelete files deleted using BeckerTools
```

BeckerTools Version 2 Plus is as easy as pointing and clicking with the mouse. **BeckerTools** takes advantage of Windows' multitasking capabilities and **BeckerTools** keeps you informed of its progress as it works.

Here are some of the things that you can easily do with **BeckerTools Version 2:**

- Launch applications - directly from **BeckerTools**
- Associate files - logically connects applications with their file types
- Backup (pack files) hard disk - saves 50% to 80% disk space - with password protection
- User levels - 3 levels for beginning, intermediate and advanced users
- Undelete - recover deleted files
- Undelete directories - recover directories in addition to individual files
- Delete files - single, groups of files or directories, including read-only files
- Duplicate diskettes - read diskette once, make multiple copies
- Edit text - built-in editor with search and replace
- Format diskettes - in any capacity supported by your drive and disk type
- Compare diskettes - in a single pass
- Wipe diskette - for maximum security
- Screen blanker - prevent CRT "burn in"
- File grouping - perform operations on files as a group
- Create a bootable system diskette
- Reliability checking - check for physical errors on disk media
- Edit files - new hex editor to edit virtually and type of file
- Dozens of other indispensable features

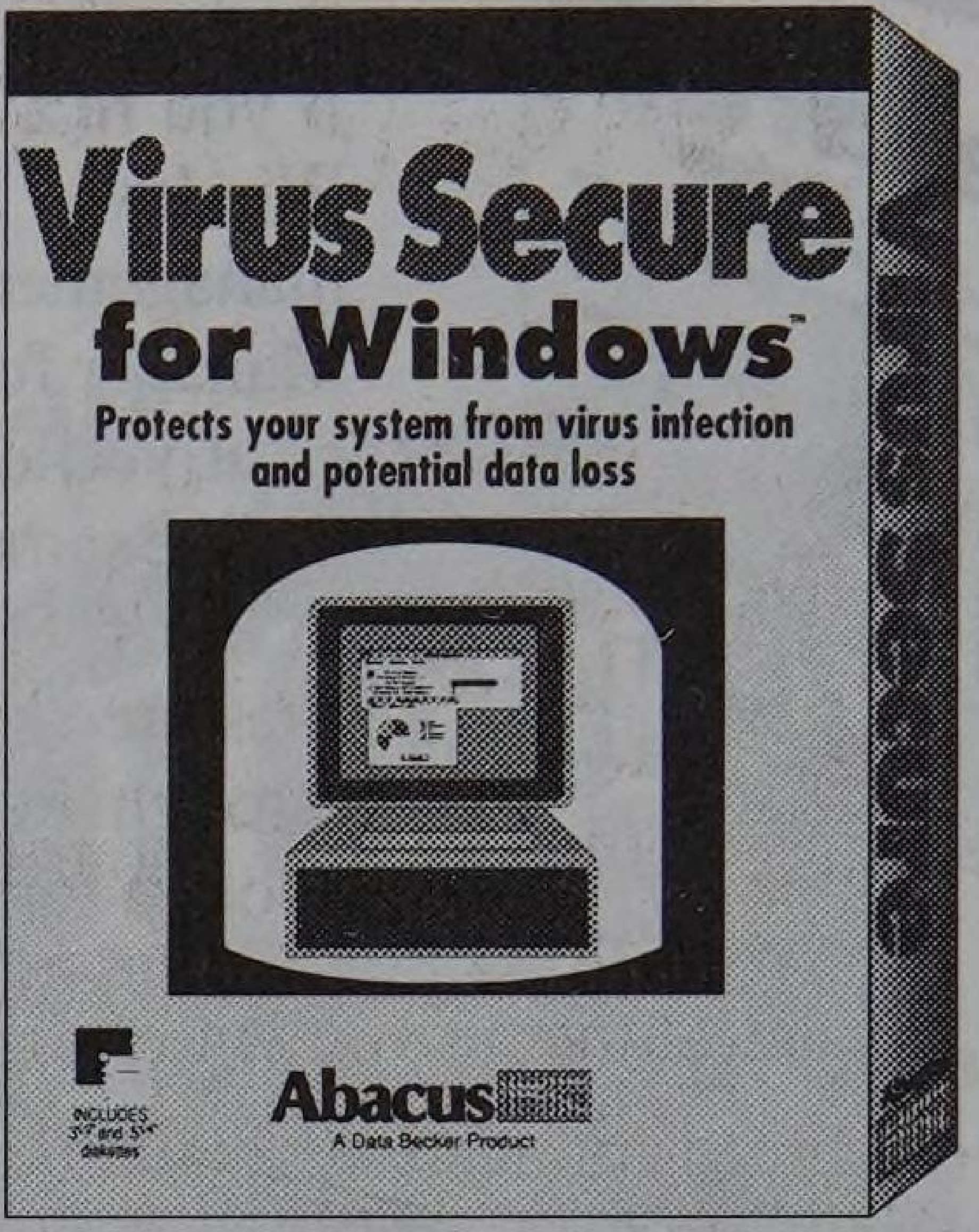
To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Microsoft Windows 3.0 Software

Virus Secure for Windows

Eliminate potential valuable data loss and minimize your risks with Virus Secure.


Virus Secure
for Windows
Protects your system from virus infection and potential data loss
Abacus
A Data Becker Product
Virus Secure

Don't give viruses the chance to ruin your data when you're using Windows. Avoid virus infection and potential valuable data loss. **Virus Secure for Windows** provides complete data protection for your hard and/or floppy disks from computer virus infection. **Virus Secure** is written by virus authority Ralf Burger (author of Computer Viruses and Data Protection). This is security that will keep your data alive and your PC operations productive.

Virus Secure can be run by both beginning and experienced users. Beginners will like the clear user interface and detailed error system. Experts will like the numerous options offered by **Virus Secure.** Beginning and advanced users will learn how to protect their diskettes and hard drives from unwanted data loss caused by viruses.

Virus Secure can detect over 200 known viruses and has built-in power for recognizing new or unknown viruses. It can also distinguish "normal" changes from "unusual" changes that viruses can make. **Virus Secure** can also be expanded. You can easily expand the recognition of viruses using a standard word processor. **Virus Secure** allows you to stay up to date as new viruses appear in the PC community.

Item #S108, ISBN 1-55755-108-1. Retail price $95.00.
System requirements: PC AT, 386 or compatible, hard drive and Windows 3.0.
Windows not included.

Author Ralf Burger

Windows is a trademark of Microsoft Corporation.

MICROSOFT WINDOWS
Version 3.0 Compatible Product

To order direct call Toll Free 1-800-451-4319

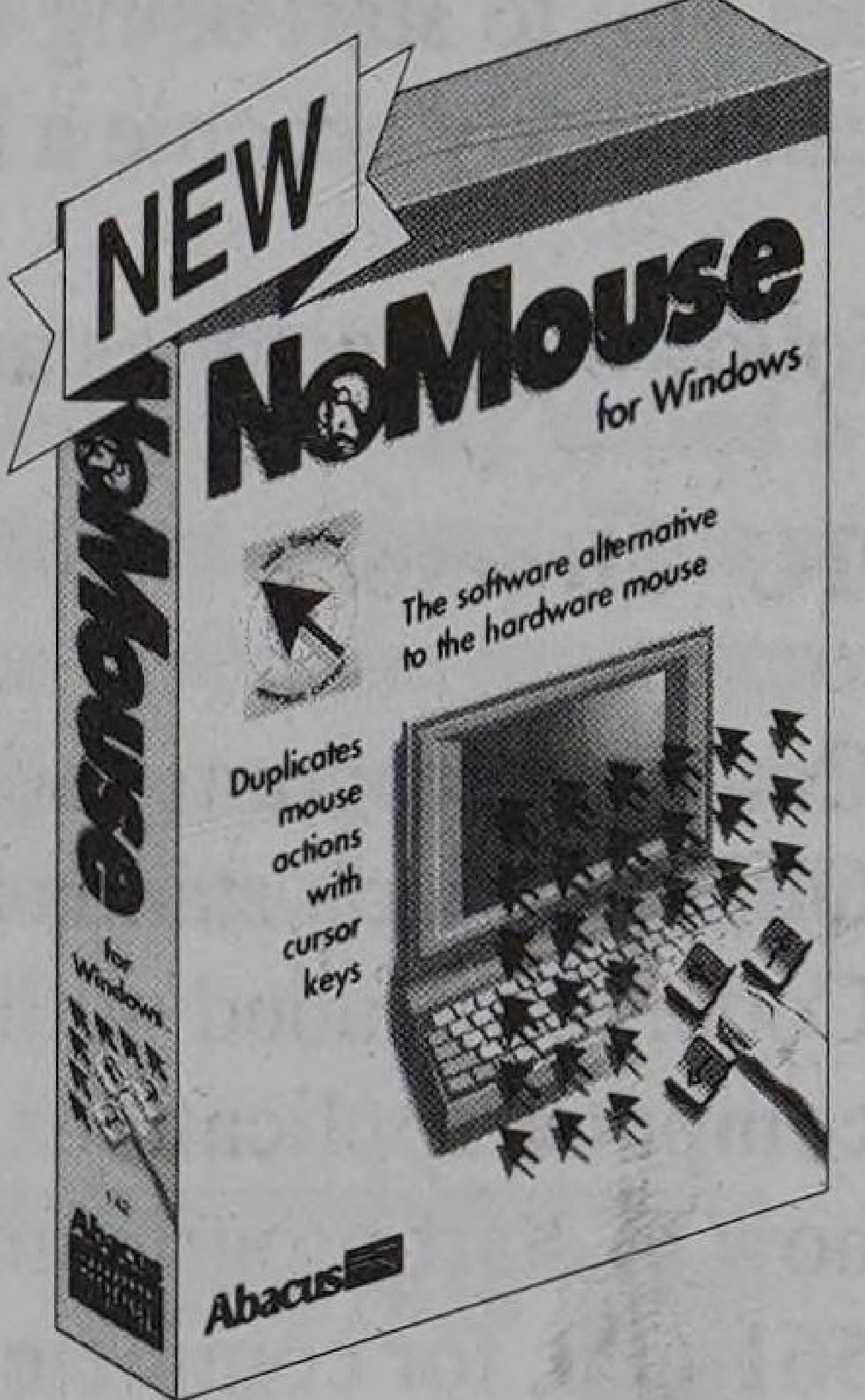
In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

Software

NoMouse™ for Windows™

For times when the best mouse is **NoMouse**™

Whether you're computing on the run or at your desk, sometimes a mouse just ties you down. **NoMouse**™ is the ideal accessory for every Windows user. **NoMouse**™ eliminates the need for a hardware mouse. It provides precise control of the mouse pointer by using the keyboard to control the movement of the cursor. **NoMouse**™ gives you all of the functions of a Windows mouse without the inconvenience. No more cords to untangle and plug in or finding a flat surface for your hardware mouse.


NEW
NoMouse for Windows
The software alternative to the hardware mouse
Duplicates mouse actions with cursor keys
Abacus

NoMouse™ gives you precision, pixel-by-pixel control of the mouse pointer. And because **NoMouse**™ is compatible and can be used in conjunction with your hardware mouse, you get exact positioning for all of your Windows applications. You can even display the mouse pointer's coordinates.

If you have trouble finding the mouse pointer on your notebook, laptop or desktop screen, you can turn on **NoMouse**™'s EasyFind. **EasyFind** eliminates the problem of locating the mouse pointer on your screen. **EasyFind** triples the size of your mouse pointer for EASY Windows computing.

- Provides exacting, pixel-by-pixel control of the mouse pointer
- Enlarges the Windows mouse pointer for easy finding
- Adds zero weight to your laptop or notebook
- Requires no extra desktop space for a mouse
- Uses only 30K of hard disk space
- Quickly configures to your computer
- Compatible with all Windows applications • Installs easily in minutes
- Instantly adjust the tracking speed and acceleration to your preferences
- Readily define the various mouse emulator "hot keys"
- Easily specify a user definable "home" screen location for the mouse pointer
- Works on laptop keyboard layouts both extended keyboard (laptops and notebooks) or numeric keyboard (desktop) layouts

NoMouse™. Includes 48pp manual and 3-1/2" diskette. ISBN 1-55755-142-1.
Suggested retail price $49.95.

Customer Comments -
"*More useable than my Microsoft BallPoint for many situations*" R.D., Canada
"*This is what every Windows user has been waiting for.*" A.L., NY
"*Very nice piece*" B.D., CA

To order direct call Toll Free 1-800-451-4319

In US and Canada add $5.00 shipping and handling. Foreign orders add $13.00 per item.
Michigan residents add 4% sales tax.

The Companion Diskette

This book is a complete reference and practical users guide to learning and using DOS 5.0. **DOS 5.0 Complete** contains numerous examples and tips for getting the most out of DOS 5.0.

The companion diskette included with this book saves you time because you don't have to type in the examples or program listings presented in the book. It also enables you to start using the many programs and tips and tricks immediately. So you can quickly become a DOS 5.0 expert.

The following directories are included on the diskette:

BASIC directory

- Contains program listings for QBasic, DOS 5.0's programming language. QBasic, which stands for Quick Beginners All Purpose Symbolic Instruction Code, is included with DOS 5.0. With QBasic you can write your own computer applications. This directory includes programs that will show you how to start programming in QBasic. See Chapter 16, Using QBasic, pages 561-600, for complete information about the files contained in this directory.

BAT directory

- Contains batch file listings that let you automate routine tasks, such as deleting backup files, searching for files, backing up changed files, and deleting the contents of a directory. See Chapter 14, Batch Files, pages 449-484, for complete information about the files contained in this directory.

DOC directory

- Contains example files that will help you quickly learn how to use the MS-DOS Editor. See Chapter 13, Using the MS-DOS Editor, pages 421-448, for complete information about the MS-DOS Editor program.

TIPTRICK directory

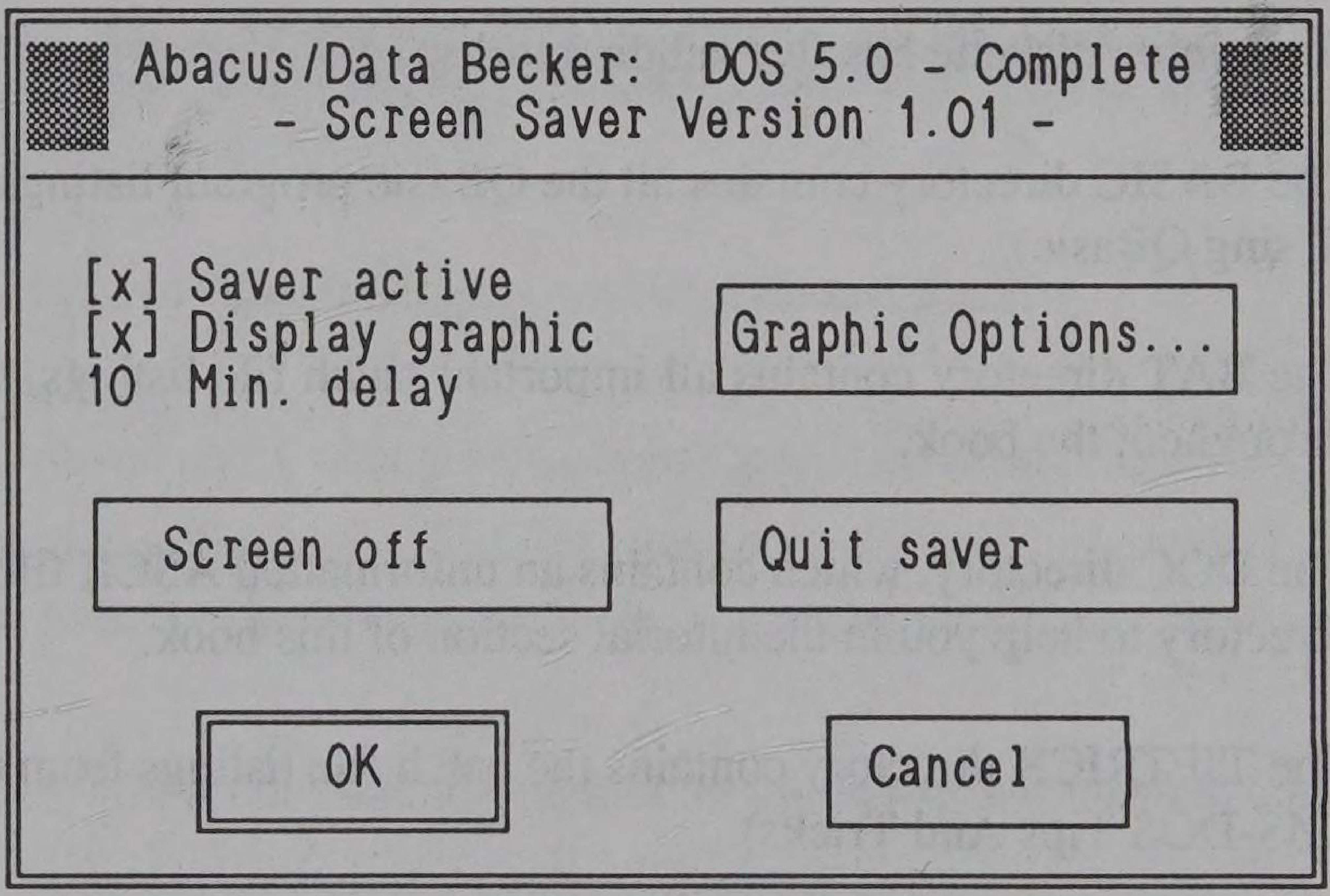
- Contains techniques that will help you use MS-DOS more effectively. Includes programs that assign DOS commands to function keys, display files in the path, change directories with a keypress, copy groups of files, and delete protected files. These tips and tricks will make you a DOS expert now. See Chapter 17, MS-DOS Tips And Tricks, pages 601-654, for complete information about the tips and trick files contained in this directory.

TOOLS directory

- Contains ready-to-run programs that speed up your work with DOS 5.0. Includes programs that configure screen colors, find files, display function key assignments, search files for a specific word, safely format diskettes (doesn't endanger hard drive), configure your printer and much more. See Appendix E, The Companion Diskette, pages 815-832 for complete information on the valuable DOS 5.0 tools included on the companion diskette.

SAVER directory

- The SAVER directory contains a DOS 5.0 screen saver utility called SAVER, which blanks the screen after a pre-determined amount of time. This protects your monitor from "screen burn", which can occur when the same screen is displayed for long periods of time. Screen savers programs alone can cost more than the price of this book/disk combination. You'll see the following when you activate SAVER:



```
Abacus/Data Becker:  DOS 5.0 - Complete
   - Screen Saver Version 1.01 -

[x] Saver active
[x] Display graphic
10  Min. delay
```

You'll find complete instructions for the screen saver in the SAVER.TXT file.

Easy Installation

The companion diskette files can easily be installed on your hard drive by using the TINSTALL.BAT program. Simply insert the companion diskette in drive A: and type TINSTALL.

See Appendix E, The Companion Diskette, pages 815-832 for installation instruction and complete information on the valuable utilities included on the companion diskette.

Book/companion diskette packages:

- Save hours of typing in source listings from the book.
- Provide complete ready-to-run batch file listings and multiple examples; help avoid printing and typing mistakes.
- This diskette contains the QBasic programs and batch files presented in the text along with many examples to make you an MS-DOS 5.0 expert now.

If you bought this book without the diskette, call us today to order our economical companion diskette and save yourself valuable time.

Abacus
5370 52nd Street SE • Grand Rapids, MI 49512
Call 1-800-451-4319

DOS 5.0 Complete Companion Diskette

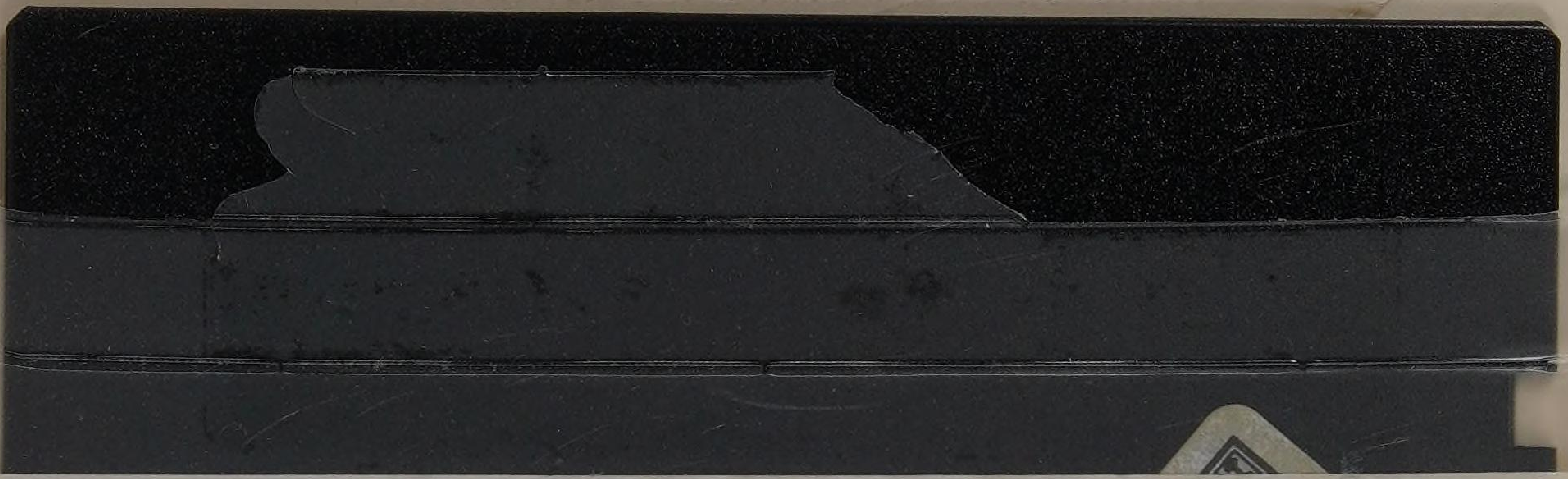
The companion diskette for DOS 5.0 Complete contains listings for the batch files and QBasic programs to save you the trouble of typing these listings in. It also contains new COM and EXE utility programs for DOS 5.0.

The companion diskette has five subdirectories:

- The BASIC directory contains all the QBasic program listings from Chapter 16 (Using QBasic).
- The BAT directory contains all important batch file listings, which are located throughout the book.
- The DOC directory, which contains an unformatted ASCII file and a PRIVATE directory to help you in the tutorial section of this book.
- The TIPTRICK directory contains the batch file listings from Chapter 17 (MS-DOS Tips And Tricks).
- The TOOLS directory contains a group of new, ready-to-run programs to streamline your sessions with DOS 5.0.
- The companion diskette allows you to start using the programs, tips, hints and tricks presented in the book immediately.

Turn Back: for information on Companion Diskette

BREAK AT SEAL AND PULL TO OPEN



4 Reasons to Register Your Product

1. **Customer Support**
   You'll be able to call our technical support personnel for assistance.
2. **Upgrade information**
   You'll receive advanced notice of new versions and special discount offers.
3. **New product Information**
   You'll be the first to hear of new books and software and special discount offers.
4. **Win prizes**
   Become eligible to enter our monthly drawing for free computer software and books.

Abacus
5370 52nd Street SE • Grand Rapids, MI 49512

PATENT 4,549,658 #41-FD PATENT D-286,469

DOS 5.0 COMPLETE

Learn to use DOS 5.0 faster, better and more productively

Not just another reference book – **DOS 5.0 Complete** is a practical user's guide to learning and using Microsoft's new DOS. It's an encyclopedia of DOS knowledge not only for the computer whiz but for the everyday user.

DOS 5.0 Complete is loaded with helpful hints for outfitting any computer with MS-DOS 5.0. From installing DOS 5.0 to using the new file, directory and storage maintenance features, you'll find many techniques and hints.

DOS 5.0 Complete has dozens of easy to follow examples. Fully explains AUTOEXEC.BAT and CONFIG.SYS. The detailed explanations make this the most authoritative DOS reference book available. The friendly, easy to understand writing style insures that even beginners will grasp the fundamentals quickly and easily. And you'll find a complete DOS command reference.

DOS 5.0 Complete includes a companion disk with dozens of powerful batch files, detailed explanations and useful tips and tricks to help you get the most out of MS-DOS 5.0.

A practical user's guide to learning and using the new MS-DOS 5.0

Topics include:

- Using DOS' new memory Management features.
- Using the improved SHELL for performing your computer housekeeping chores.
- Using the new DOSKEY utility for faster command line editing and macro power.
- Using EDIT, the new full-screen editor.
- Using QBASIC, DOS' new BASIC programming language.
- Complete DOS command reference.
- ScreenSaver included on companion disk.

Includes ready-to-use companion diskette

ISBN 1-55755-109-X

$34.95 USA

9 781557 551092

Computer Book Category
IBM/PC: DOS Programming

Abacus
5370 52nd Street SE • Grand Rapids, MI 49512